

بدء العمل مع Microsoft Silverlight 4

تأليف

عبدالرحمن الغبان

مراجعة وتدقيق

وجدان المشهري



عزيزي المبرمج الوسيم / عزيزتي المبرمجة الجميلة :

انتهائك من هذا الكتاب لا يعني بالضرورة أن تصبح أفلاطون في Silverlight و لكنك ستحصل على أهم الأساسيات و المعرفة التي تحتاجها لبدء العمل مع Silverlight .



أ : قبل بدء العمل مع Silverlight :

قبل أن تبدأ في قراءة الفصل القادم أود أن أضع مجموعة من التعليمات و الأمور التي يجب أن تعرفها عن Silverlight ، من الجيد جدا أن تعرف كيفية عمل Silverlight و دورة حياة هذا النوع من التطبيقات و أدوات التطوير ، إذا كنت فعلا تمتلك هذه المعلومات سابقا ، يمكنك تجاهلها ، أما إذا كانت هذه أول مرة تقرأ فيها كتاب عن Silverlight فأرجوك لا تهمل هذه المقدمة .



تثبيت أدوات التطوير الخاصة بـ Silverlight :

لا تأتي حزم و أدوات تطوير Silverlight مباشرة مع Visual Studio 2010 ، بل يجب عليك تثبيتها بشكل مستقل عن Visual Studio 2010 ، ستتعرف في هذا الفصل إلى الأدوات التي ستحتاجها لبدء العمل مع Silverlight مع الأخذ بعين الاعتبار أني أفترض أنك قد قمت بتثبيت نسخة من بيئة التطوير Visual studio 2010 مسبقا ، إذا كنت أحد القابعين في الإصدارات الأخيرة من Visual Studio 2010 فأنا أحد الأشخاص الذين على استعداد لتقبيل يدك الجميلة للانتقال إلى 2010 بأسرع وقت ممكن ، من جهة أخرى لا يمكنك متابعة هذه الكتاب إذا كنت لا تملك نسخة من بيئة التطوير visual studio 2010 .

- ## تثبيت Silverlight 4 Tools for Visual Studio 2010 :

بعد أن تتأكد من تثبيتك لأي من نسخ visual studio 2010 يجب عليك تثبيت نسخة من Silverlight 4 Tools for visual studio 2010 ، هذه الحالة فقط إذا كانت النسخة التي معك ليست SP1 ، فمن المفترض أنك إذا قمت بتثبيت النسخة SP1 ستجد Silverlight قد ثبت بالفعل أثناء عملية تثبيت visual studio 2010 . أولا قم بعملية التحميل لهذه الأدوات من Silverlight 4 Tools for visual studio بعد الانتهاء من عملية التحميل ابدأ بتثبيتها بطريقة عادية مع ملاحظة أن تكون قد أغلقت visual studio أولا .



تجد الأدوات السابقة على الرابط التالي :
http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=18149

- ## تثبيت أداة Silverlight developer Run Time :

كل ما يجب أن تعرفه عن هذه الأداة هو أنك لن تستطيع أن تقوم بتشغيل أي تطبيق لـ Silverlight بدون وجود Silverlight developer Run Time مثبته على جهازك ، إذا حاولت أن تقوم بتجربة أي تطبيق دون تثبيت هذه الأداة فسيخبرك visual studio بشكل مباشر عند محاولتك لتشغيل التطبيق بأن هذه الأداة غير موجودة . بعد أن تنتهي من تحميل الأداة Silverlight developer run time أغلق Visual studio و قم بتثبيت هذه الأداة .



تجد الأداة السابقة على الرابط التالي :
http://download.microsoft.com/download/C/5/8/C5877489-8022-42A6-8456-D05C2A6C8DFB/runtime/Silverlight_Developer.exe

• Microsoft Expression Blend 4 :

هذه ليست أداة لابد من توافرها لكي تعمل مع Silverlight و لكن Blend برنامج تصميم سيغير كليا من طريقة تصميمك لتطبيقات Silverlight ، سيوفر عليك جهدا ووقتا كبيرا من الممكن أن تقضيه في تعلم الأمور الأكثر تأثير في طريقة كتابتك لتطبيقات Silverlight بدلا من إضاعة الوقت في تعلم XAML .



تجد البرنامج السابق على الرابط التالي :
http://www.microsoft.com/expression/products/blend_overview.aspx

• تثبيت Silverlight toolkit :

على الرغم من الأدوات الأكثر من رائعة التي تقدمها Silverlight إلا أنك ستنبهر حقا بمجموعة أكبر من الأدوات التي تأتي مع toolkit مثل خريطة Bing و أدوات الرسم البياني و العديد من الثيمات المختلفة في Silverlight . يمكنك تحميل toolkit من هنا .



تجد الأداة السابقة على الرابط التالي :
http://silverlight.codeplex.com/

• تثبيت Deep Zoom Composer :

أخيرا أداة Deep Zoom Composer و هي من أكثر الأدوات و أروعها استخداما في عالم Silverlight ، حيث تستخدم هذه الأخيرة في عرض مجموعة من الصور على شكل تكبير و تصغير ، ستتضح الفكرة لاحقا ، عند الحديث عن هذه الأداة .

يمكنك تحميل أداة Deep Zoom Composer من هنا .





تجد البرنامج السابق على الرابط التالي :

http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24819

القرص المرفق مع الكتاب:

يحتوي القرص المرفق مع هذا الكتاب على مجلدات تمثل فصول الكتاب (رقم المجلد يحمل رقم الفصل)، يحوي كل مجلد على ملف الشفرات المصدرية للمثال الذي تم تطبيقه في الفصل.



دورة حياة تطبيقات Silverlight

لا يمكن أبدا تجاهل دورة حياة أي نوع من التطبيقات التي تقوم بتطويرها فمن المهم جدا أن تفهم كيف تدار الأمور خلف الكواليس ، وذلك في حال واجهتك المتاعب تستطيع أن تتصرف بحكمة و ذكاء ، في هذه الفقرة سأتحدث عن دورة حياة التطبيقات الخاصة بـ Silverlight ، يمكنك تجاهل هذا الفصل إذا كنت على معرفة سابقة بهذه المعلومات حيث يمكن اعتبارها معلومات عامة و لكن مهمة جدا لكل مبتدئ في عالم تطبيقات Silverlight.

حسنا لندخل في صلب الموضوع ، في هذه النقطة سنحاول شرح دورة حياة تطبيقات Silverlight ، هناك نقطة يجب أن تعرفها عند الحديث عن تطبيقات Silverlight فهي تطبيقات شبيهة جدا بتطبيقات سطح المكتب و في نفس الوقت بعيده كليا عن تطبيقات سطح المكتب ، أعلم جيدا أنك ستنعتني بالمجنون بعد الجملة السابقة ، و لكن هذا هو الواقع ☺ ، و هنا يأتي دور الحديث عن دورة حياة تطبيقات Silverlight

تبدأ تطبيقات Silverlight حياتها منذ اللحظة التي تضع فيها تطبيقك على صفحة asp.net ، يمكن توضيح تفاصيل ذلك بالشكل التالي :



١. يقوم المستخدم بطلب صفحة تحتوي على تطبيق Silverlight .
٢. يتم التأكد من أن المستخدم قد قام بتثبيت نسخة من Silverlight run time متوافقة أو أعلى من النسخة التي قام المطور بتطوير تطبيقه بها .
٣. يتم تحميل تطبيق Silverlight إلى جهاز المستخدم ، تطبيق Silverlight ينتهي بامتداد xap. وهذا الملف يتم إنشائه في مسار معين تتعهد مايكروسوفت بأن يكون هذا المسار مجهول حيث يتم حفظ تطبيق xap. داخل هذا المجلد .
٤. لاحظ هنا أن ما يتم تحميله هو فقط المكونات الرئيسية للتطبيق مثل الصور و خلافه ولا يتم تحميل مثلا خدمات الويب .
٥. بعد تحميل التطبيق إلى جهاز العميل تتولى هنا CLR عملية تجميع التطبيق و تنفيذه ، عند الانتهاء من التحميل تطلق الأخيرة حدث startup و الذي بدوره يبدأ بعرض التطبيق على المستخدم .

تحدث هذه العملية مرة واحدة فقط في المرة الأولى التي يتم فيها فتح صفحة Silverlight، في المرة القادمة فقط يتم معالجة الأحداث المختلفة أو إن صح التعبير معالجة أحداث التطبيقات .

تطبيقات Silverlight و ASP.net :

حسنا هذه نقطة شائكة نوعا ما ، يجب أن تعلم أمرا مهما جدا جدا عند الحديث عن تطبيقات Silverlight فهي تختلف جذريا عن تطبيقات asp.net ، إذا أمعنت النظر في تطبيقات asp.net فهي تطبيقات يتم تنفيذها على جهاز الخادم أي من جهة الـ Server بينما تطبيقات Silverlight يتم تنفيذها على جهاز العميل كما رأينا في دورة حياة التطبيق في الأعلى .

في تطبيقات asp.net يتم توليد شفرات CSS و HTML و JS ، بينما في تطبيقات Silverlight لا يتم توليد أي شيء سوى صفحة واحدة تحتوي على شفرة JS و التي تكون مسؤولة عن استضافة تطبيق Silverlight .

أخيرا لنتحدث قليلا عن ما يحدث عند التحميل بشكل أكثر تفصيلا ، هناك سبعة أحداث تطلق بشكل مستمر عند تحميل أي تطبيق Silverlight و هي بالشكل التالي:

١. كل شيء يبدأ من ملف app.xaml ، حيث أنه في داخل هذا الملف يتم تحديد أي صفحة من صفحات Silverlight ستعمل أولا .
٢. لدينا الحدث application_startup و فيه تستطيع أن تقوم بمجموعة تعليمات قبل عرض التطبيق على المستخدم .



٣. في الحدث السابق يتم التوجه لأول rootVisual حيث كل صفحة Silverlight يجب أن تحتوي على rootVisaul .

٤. يتم إطلاق الحدث sizeChange .

٥. يتم إطلاق الحدث LayoutUpdated .

٦. يتم إطلاق الحدث GetFocus .

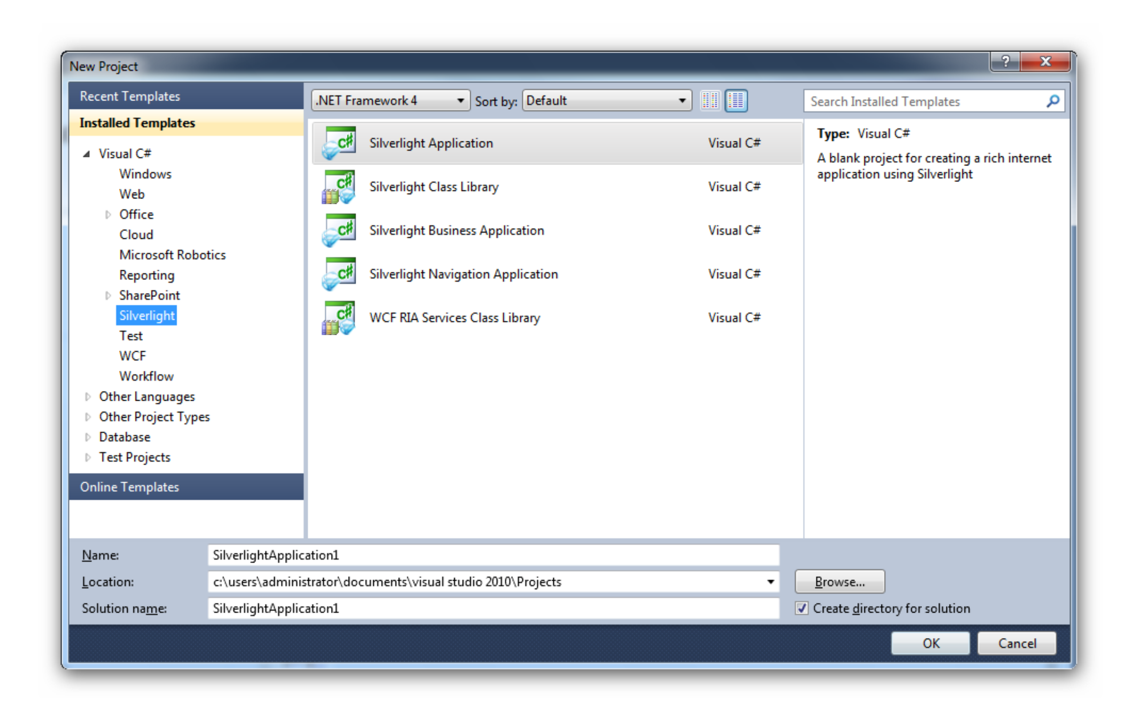
٧. إذا أغلق المستخدم الصفحة يتم إطلاق الحدث application_Exit .

مرحبا بالعالم من Silverlight

قبل أن ننتقل إلى الأجزاء الجديدة في Silverlight دعنا نأخذ جولة سريعة على بيئة التطوير الخاصة بـ Silverlight ، سأعتبر هنا أنك قمت بتثبيت نسخة من أدوات التطوير المختلفة كما تحدثنا سابقا ، لذلك لنبدأ بالعمل .

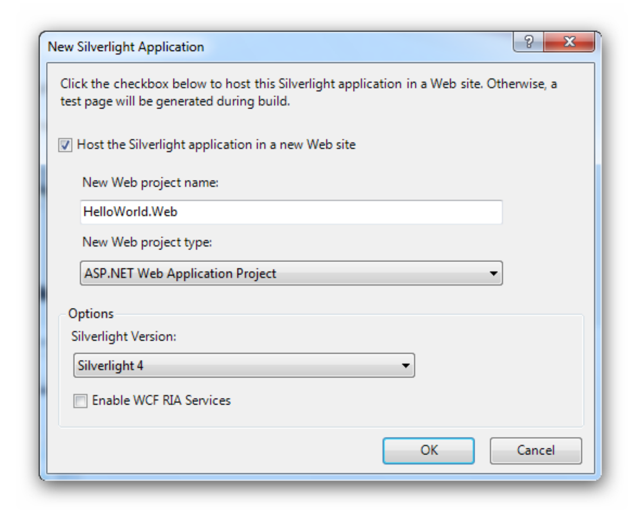
ابدأ مشروع جديد باستخدام Silverlight :

افتح Visual studio و اختر من file > new > project ، أسفل الخيارات ستري عنصر جديد باسم Silverlight حدد هذا العنصر و من ثم اختر الخيار الأول Silverlight Application ، قم بتغيير اسم المشروع إلى HelloWorld .







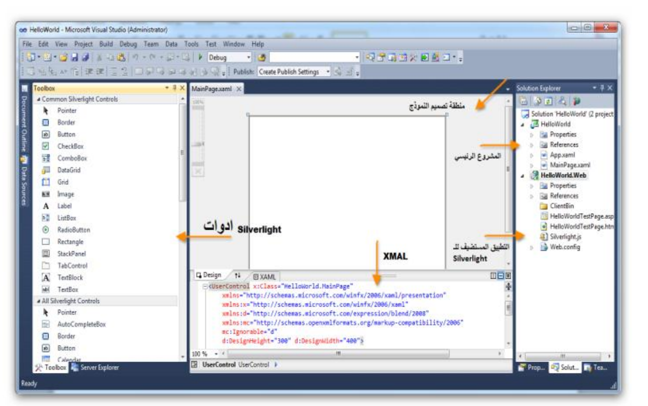


بعد أن تختار مشروع Silverlight ستخرج لك نافذة أخرى تطلب منك بعض المعلومات (انظر الصورة) ، كما تحدثنا في دورة حياة التطبيق ستحتاج إلى مضيف لتطبيق Silverlight و هو مشروع جديد من نوع تطبيق ويب يحتوي على صفحتين الأولى asp.net، الثانية HTML ، أيضا تمنحك هذه النافذة خيار تحديد أي نسخة من Silverlight يستهدفها مشروعك ، بشكل تلقائي ستكون النسخة الأخيرة .

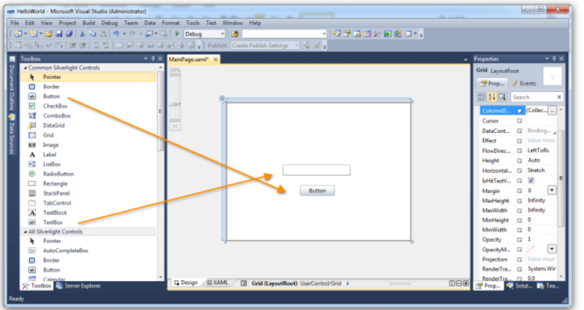
أخيرا لديك الخيار Enable WCF RIA Services و هو خيار يتيح لك تشغيل تطبيقات Silverlight من خارج المتصفح ، سنتطرق لهذا الأمر لاحقا (في الواقع بشكل ممل لاحقا ☺) .

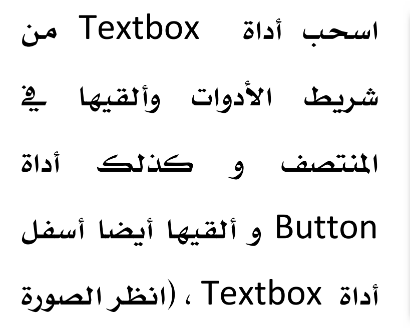
اترك كل شيء كما هو حاليا و انقر على OK .





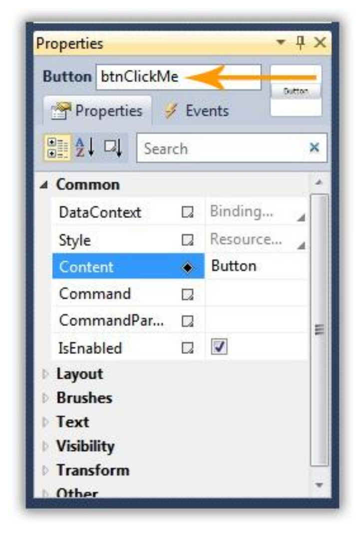
كما ترى في النافذة على اليسار لديك خمسة أجزاء تقريبا عندما نتحدث عن تقسيم شاشة العمل ، لدينا منطقة XAML و من خلالها يتم تصميم التطبيق عن طريق كتابة الأكواد ، بينما لدينا منطقة التصميم و يتم من خلالها تصميم النموذج من خلال السحب و الإلقاء للأدوات المختلفة ، لدينا على الجهة اليسرى من النموذج الأدوات الأساسية لتطبيقات Silverlight بينما على يمين الشاشة لدينا مشروعين الأول تطبيق Silverlight و الثاني تطبيق Web أو المستضيف و سنتحدث عن هذا التطبيق لاحقا و لكن دعنا نبدأ في تطبيق Silverlight أولا .





اسحب أداة Textbox من شريط الأدوات وألقيها في المنتصف و كذلك أداة Button و ألقيها أيضا أسفل أداة Textbox ، (انظر الصورة على اليمين) ، إذا كنت لا ترى نافذة الخصائص يمكنك إظهارها من خلال view > window preppies .





حدد أداة Textbox و قم بتغيير الخاصية name في أعلى الشريط إلى txtname وقم بنفس الأمر مع الأداة Button وقم بتغيير الخاصية name إلى btnClickMe (انظر الصورة على اليسار) .

لننظر إلى شفرة XAML التي تم توليدها حتى هذه اللحظة و لنشرحها بشيء من التفصيل :

```
<UserControl x:Class="HelloWorld.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">

    <Grid x:Name="LayoutRoot" Background="White">
        <Button Content="Button" Margin="0,0,189,241" Name="button1" Height="23"
                VerticalAlignment="Bottom" HorizontalAlignment="Right" Width="75" />
        <TextBox Height="23" HorizontalAlignment="Left" Margin="10,10,0,0"
                 Name="textBox1" VerticalAlignment="Top" Width="120" />
    </Grid>
</UserControl>
```

فضاءات الأسماء

الشفرة الناتجة عن عملية السحب والإلقاء

الفضاءات أداة من أدوات Silverlight تكون موجودة داخل فضاء أسماء يتم تعريفه أعلى ملف الشاشة ، نفس الأمر الذي يحدث عندما تود استخدام I/O مثلا فعليك



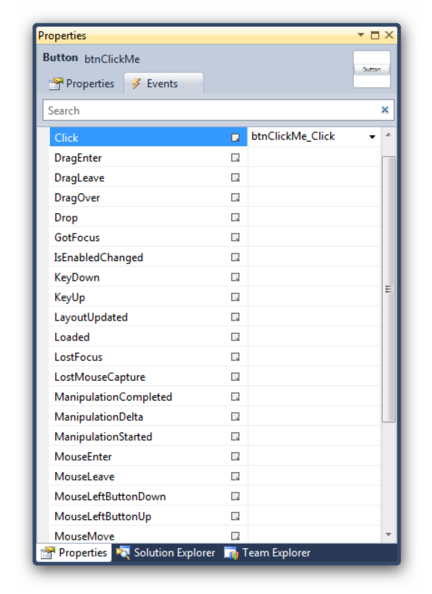
استدعاء فضاء الأسماء system.IO أعلى الشفرة في سي شارب ، أما الجزء الثاني من الشفرة فهي الشفرة التي تم توليدها عند عملية السحب و الإلقاء ، يجب أن تلاحظ هنا أن LayoutRoot شيء أساسي في كل نماذج Silverlight و كل الأدوات تكون داخل LayoutRoot ، لاحظ هنا أن هذه الأخيرة ليست سوى Grid سنتحدث عنها لاحقا و كما تعرفنا سابقا فإن هذا الجزء هو الجزء الأول الذي يتم بناءه عند تحميل التطبيق (راجع نقطة دورة حياة تطبيقات Silverlight) .

عد إلى نافذة التصميم و انقر نقرا مزدوجا على Button المسمى btnClickMe ، نتيجة لذلك ستفتح نافذة الشفرة و قد قام Visual studio بشكل آلي بتوليد الحدث Click و هنا يمكنك كتابة شفرة سي شارب ، اكتب مثلا :

```
1:  txtName.Text = " Hellow World ";
```

عد إلى صفحة XAML، لاحظ الحدث الذي تم توليده للعنصر butClickMe ، كما في الشفرة :

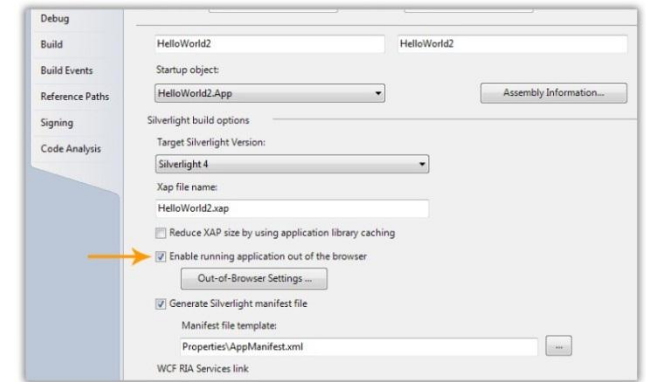
```
1:  <Button Content="Button" Height="23"
2:  HorizontalAlignment="Left"
3:  Margin="160,179,0,0" Name="btnClickMe"
4:  VerticalAlignment="Top" Width="75"
5:  Click="btnClickMe_Click" />
```



هناك العديد من الأحداث التي يمكنك التعامل معها من خلال Silverlight ، من نافذة الخصائص حدد التبويب Events و سترى ما يسر نظرك بكل تأكيد (انظر الصورة على اليمين) أحداث لم تكن تحلم بها سواء في تطبيقات asp.net أو تطبيق WPF.

حسنا لنقم بتشغيل التطبيق ، الآن أنقر على زر F5 أو على Start Debugging من شريط Debug للبدء بتنفيذ التطبيق ، سوف تفتح شاشة المتصفح و بها تطبيق Silverlight الذي قمت بتطويره قبل لحظات ، إذا نقرت على الـ Button ستخرج لك رسالة Hello world داخل TextBox .

أغلق المتصفح وعد مجددا إلى بيئة التطوير و حدد التطبيق Silverlight من نافذة Solution Explore ثم انقر بزر الفأرة الأيمن و اختر الأمر Properties ، بشكل تلقائي سوف يحدد Visual studio التبويب Silverlight انزل إلى الأسفل قليلا وقم بتحديد الأمر Enable running application out of browser (انظر الصورة على اليمين)



تأكد من تحديدك لهذا الخيار ثم أعد تشغيل التطبيق مجددا بالضغط على F5، هل أصبت بذهول !! نعم هذا صحيح عزيزي لقد أصبح تطبيقك يعمل بشكل كامل كتطبيق سطح مكتب و ليس تطبيق ويب !! و هذا الأمر يعود الفضل فيه إلى دورة حياة تطبيقات Silverlight كما تحدثنا سابقا ، هناك فصل كامل يتحدث عن هذا الأمر فلا تستعجل أرجوك ، فقط امنحني القليل من صبرك و ووقتك .

إلى هنا أكون انتهيت من المقدمة ، تطرقت خلالها إلى أدوات التطوير و بيئة العمل و تقسيم العمل و رأينا كيف تؤثر عملية السحب و الإلقاء على شفرة XAML و تعرفنا على دورة حياة تطبيقات Silverlight و رأينا في الجزء الأخير كيف أن دورة الحياة تؤثر بشكل مباشر على طريقة عمل تطبيقات Silverlight .

١

الجزء الأول

البداية مع Silverlight

في هذا الجزء ستتعرف على اساسيات التصميم في برنامج Microsoft Blend والتعامل مع animation ثم سنتعرف على أدوات Silverlight القياسية وأهم خصائصها وأحداثها .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي:

- التعرف على Microsoft Blend
- التعرف على خصائص وأحداث الأدوات القياسية في تطبيقات Silverlight



الفصل الأول :

تصميم تطبيقات Silverlight

حسنا لست فخورا بما سأفعله في الصفحات القادمة ، ولكن جرت العادة عند الحديث عن تطوير تطبيقات Silverlight أن يبدأ الكاتب مباشرة مع أدوات التحكم الموجودة في Silverlight ، مثلا Textbox و Button و غيرها ، لكن مع الأسف الشديد سأخالف الشريعة هذه المرة و لن أبدأ بالحديث عن الأدوات ، و لكني سأتحدث عن التصميم فكل شيء في Silverlight عبارة عن تصميم .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

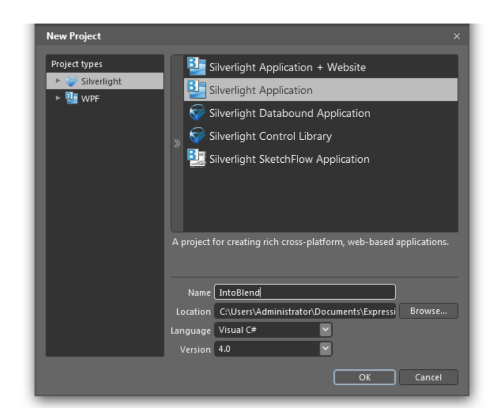
- استخدام البرنامج العظيم Microsoft Blend .
- التعامل مع animation .
- تعديل Style الأدوات الموجودة من قبل .

أرجو أن تكون قد قمت بتثبيت النسخة الرابعة من Blend كما تطرقنا في المقدمة .

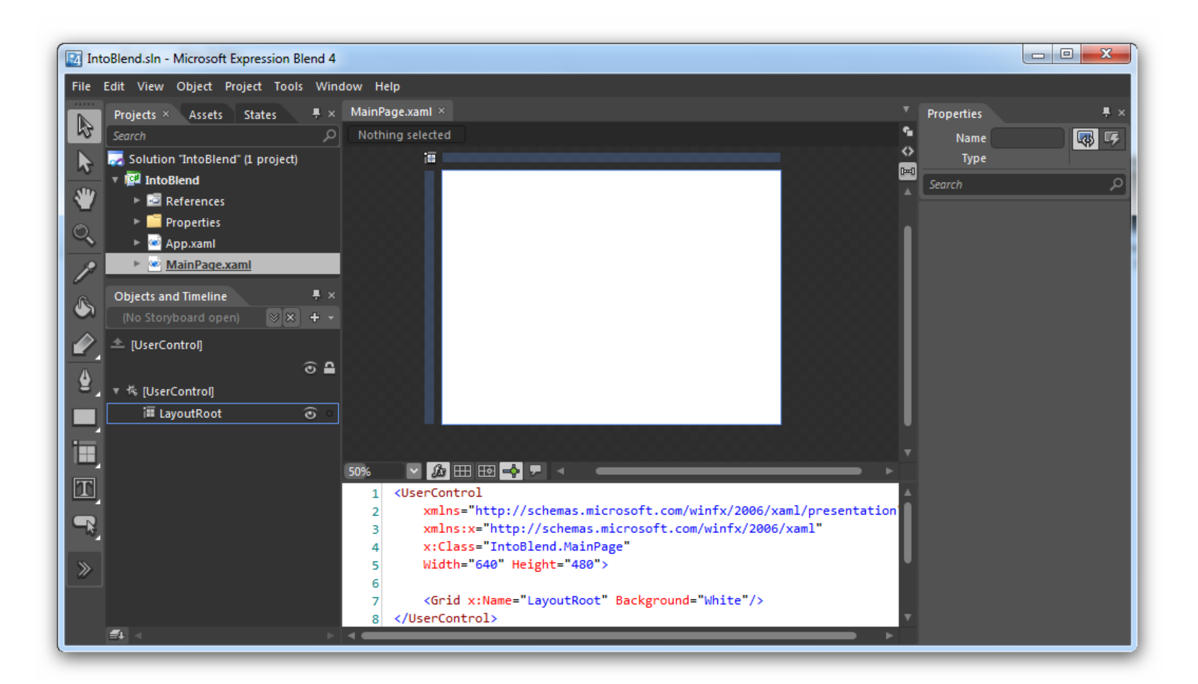
التعرف على Microsoft Blend :

في بداية ظهور Silverlight ، كان واجبا عليك أن تتعلم XAML حتى تعرف كيف يتم إنشاء التطبيقات ، بل لم تكن تستطيع أن تفعل أي شيء إذا كنت لا تعرف XAML ، و هنا قامت Microsoft لعلمها مسبقا طبعا بأن هذه اللغة الجديدة ستكون عائقا أمام المطورين للانتقال إلى WPF أو Silverlight ، بتطوير Microsoft Blend ، حيث يمكنك أن تقول بأن Microsoft Blend هو المكان المناسب لتصميم تطبيقات Silverlight و أيضا WPF .

حسنا لنبدأ بالعمل ، افتح Blend و من نافذة Welcome حدد الخيار new project ثم اختر Silverlight Project قم بتغيير اسم المشروع إلى IntoBlend حدد اللغة سي شارب و اختر Silverlight Application . حدد Silverlight 4 كنسخة المشروع ثم انقر على OK (انظر الصورة على اليمين) .



في الحقيقة لا يوجد شيء جديد يختلف في واجهة Blend عن visual Studio ، سوى نافذتين تقريبا أو ثلاثة ، لذلك لن أتطرق لطريقة استخدام blend فالأمر أشبه باستخدام برنامج الرسام ، أما عن النافذتين المختلفتين فهما بالشكل التالي أولا Object and TimeLine(OATL) و هنا يمكنك رؤية كل الأدوات التي تقوم برسمها على النموذج داخل قائمة منظمة بشكل جيد بحيث يسهل التعامل معها أثناء عملية التصميم . أيضا من نفس هذه النافذة يمكنك إنشاء ما يعرف بـ Storyboard وهو تحريك الأدوات خلال فترة زمنية معينة ، كما سنرى في هذا الفصل لاحقا .

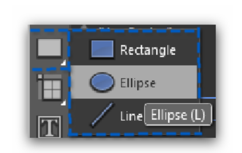




النافذة الثانية هي نافذة status و هي تقوم بعمل نفس الأمر الذي تقوم بعمله Storyboard و أيضا سنتطرق لهذه النافذة نهاية هذا الفصل . أما ما بقى من أشرطة فأعتقد أنه من السهل جدا أن تستنتج بنفسك ماذا تفعل ، الأمر فقط اختلاف بسيط في الأسماء عما هو عليه في visual studio.

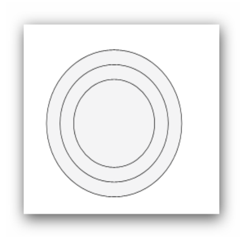
النافذة object and timeline و التعامل مع الألوان في Blend و تأثير ثلاثي الأبعاد :

في حقيقة الأمر كل التأثيرات ثلاثية الأبعاد تقريبا ما هي إلا خدع بصرية في طريقة التلوين و رسم الكائنات ، سنحتاج في هذا المثال إلى ثلاثة أدوات من النوع Ellipse ، ستجد هذه الأداة في الجهة اليسرى الأداة الرابعة من الأسفل (انظر الصورة على اليمين) ، إذا كنت لا تراها انقر بزر الفأرة الأيمن على الأداة و ستجدها في القائمة



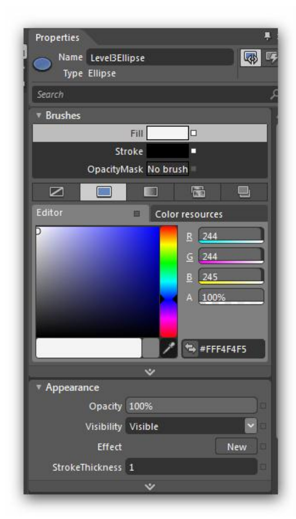
حددها ثم انتقل إلى النموذج الأبيض الذي في منطقة التصميم و ارسم Ellipse مرة واحدة في أي مكان بشكل عشوائي . بعد أن تنتهي من الرسم انتقل إلى نافذة الخصائص (إذا كانت نافذة الخصائص غير موجودة يمكنك عرضها من windows > preppies) ، في نافذة الخصائص حدد المجموعة Layout ثم قم بتغيير العرض والطول إلى ٢٥٠ ، أيضا أعد تسمية Ellipse إلى BeasdEllipse ، من نافذة object and TimeLine حدد BeasdEllipse و انقر بزر الفأرة الأيمن و اختر Copy ثم مجددا اختر Paste ستضاف أداة جديدة للنموذج الذي

أمامك تحمل الاسم BeasdEllipse _Copy قم بتغيير اسم الأداة إلى Level2Ellipse ، هذه المرة من نافذة التصميم كرر نفس العملية حدد Level2Ellipseبالفأرة باستخدام Ctrl + C و ألصقها من جديد و غيّر اسم الأداة الجديدة إلى Level3Ellipse .



تأكد بأنك قمت بتحديد الأداة Level3Ellipse و قم بتغيير حجم الأداة من المجموعة Layout إلى ١٥٠ طولا و عرضا ، قم بتحديد الأداة الثانية Level2Ellipse ثم قم بتغيير الحجم إلى ٢٠٠ طولا و عرضا ، اترك الأداة BeasdEllipse في شأنها . عد مجددا إلى نافذة (OATL) Object and TimeLine و حدد جميع الأدوات الثلاثة (لا تقم بتحديد LayoutRoot)

ثم انقر بزر الفأرة الأيمن و حدد الخيار align و من ثم horizontal center لكي تكون جميع الأدوات متطابقة فوق بعضها بالمنتصف ، استمر بتحديد جميع الأدوات انقر بزر الفأرة الأيمن و حدد الخيار Group into ثم اختر Grid ما قمنا بعمله الآن هو تجميع هذه الأدوات كلها داخل Grid واحد ، و هذا الأمر سيسهل عليك كثيرا التحكم بجميع هذه الأدوات خصوصا عند الحديث عن تحريكها من مكان إلى أخر ، من المفترض الآن أن تكون هذه الأدوات جميعها فوق بعض و متناسقة بطول واحد .



لننتقل الآن إلى عملية التلوين ، في البداية سنتعامل مع العنصر Level3Ellipse حدد العنصر من نافذة OATL أو من منطقة التصميم بأي الطريقتين تريد و انتقل إلى نافذة الخصائص ، حسنا على عكس visual studio و الذي يتعامل مع الألوان في أكثر من منطقة ، يأتي blend ليجمع كل الألوان التي تستطيع تغييرها لأي أداة كانت في أعلى شريط الخصائص.

ينقسم شريط الألوان إلى ثلاثة أقسام أولا المنطقة المراد التعامل معها و تسمى Brushes مثل الخلفية و الحدود و لون الخط .. الخ ، أما المنطقة الثانية و هي طريقة التلوين و لها أربعة أشكال إما أن تكون شفافة (no Brushes) أو مصمتة بلون واحد (Solid Brushes Color) أو متدرجة الألوان (Gradient Color) أو الحصول على الألوان من مصدر في المشروع ، كما سنرى لاحقا .

هنا حدد الخيار Solid Brushes Color ثم اختر اللون الأسود ، حدد الآن Level2Ellipse و اختر هذه gradient Color أو الألوان المتدرجة ، عند تحديدك لهذا النوع من التلوين سترى في الأسفل قليلا شريط التعبئة و يجب أن يحتوي على مسارين على الأقل ، أيضا يمكنك أن تحدد أكثر من تدرج بالنقر على هذا الشريط ، أضف لون ثالث بالنقر في أي مكان و حدد اللون الرصاصي أو اكتب هذا الرقم في خانة اللون #FFC3C3C3 ، من الأسفل على اليمين حجم التعبئة حدده ٥٠٪ ،



لننتقل الآن إلى الأداة الأولى BeasdEllipse قم بتغيير عرض الأداة إلى ٢٠٠ و مجددا غير الألوان إلى gradient Color ، هذه المرة انزل قليلا إلى المجموعة Appearance و هي مجموعة تتعامل مع التأثيرات ، من الخيار Effect اضغط على new و حدد الخيار DropShadowEffect ، الآن سوف يصبح لديك خيارات جديدة للتعامل مع هذا Effect الجديد ،



إذا كنت لا ترى هذه الخيارات يمكنك إظهارها من خلال نقر الاسم الذي بجانب كلمة effect ، قم بتغيير الخاصية BluRadius إلى ١٠٠ و كذلك الخاصية Opacity إلى ١٠٠٪ ، و اجعل الخاصية ShadowDepth 0 ، ليصبح لديك في نهاية المطاف شكل وعاء الحجر الأسود الكريم كما هو على جدار الكعبة المشرفة (الصورة على اليمين).

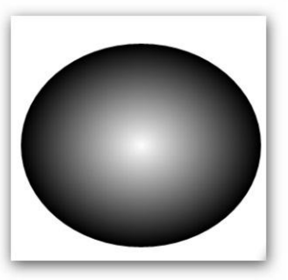
بقي لدينا الآن رسم الحجر الأسود ، لن نقوم فعلا برسم الحجر الأسود و لكن ما سنقوم بعمله هو انعكاس أشعة الشمس على الحجر الأسود ، الهدف من هذا الأمر هو التعرف على animation في Silverlight و التحكم بـ Storyboard و التعامل مع status إذا لنبدأ .



لعمل هذا النوع من الظل سنقوم أولا برسم أداة جديدة من النوع Ellipse ، ارسم الأداة بعيدا عن الرسمة السابقة و حدد العرض بـ ١٤٣ والارتفاع بـ ١٢١ ، من نافذة الألوان حدد نظام التلوين gradient Color ، اجعل بداية اللون من الجهة اليمني باللون الأسود و في النهاية بالجهة اليسرى أبيض ، ثم قم بتغيير نظام توزيع الألوان إلى Radial gradient و ذلك من خلال تحديد أسفل شريط التعبئة بالجهة اليمنى (انظر الصورة على اليمين



) . يمكنك أيضا استخدام أداة gradient tools من شريط الأدوات (الشريط الذي اخترت منه Ellipse) لجعل الدائرة تبدو بالشكل الذي على اليسار و ذلك بتحريكها من الأعلى و الأسفل و المنتصف ، ليس الأمر غاية في التعقيد بحيث يحتاج إلى شرح .



الخطوة التالية هي الانتقال إلى عملية تحريك هذه الصورة داخل وعاء الحجر بحيث تبدو كأن أشعة الشمس تنعكس على الحجر من الداخل . في البداية اختر السهم الأسود من أعلى شريط الأدوات حتى تتخلص من هذه الأداة gradient tools ثم اسحب الدائرة إلى منتصف المنطقة السوداء في الوعاء ، ليصبح الأمر كما في الشكل على اليمين .

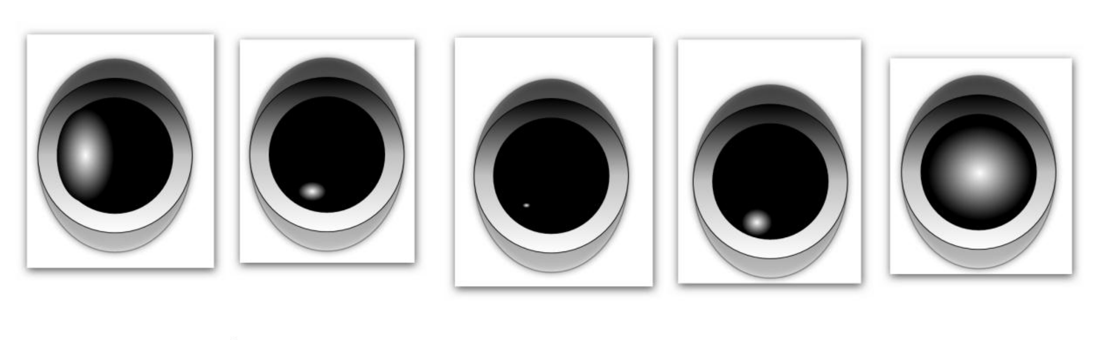




من النافذة OATL ستجد زر زائد في الأعلى انقر عليه ، ستظهر أمامك نافذة إنشاء Storyboard اجعل الاسم كما هو لا مشكلة ، بعد أن تنقر على موافق سترى زر أحمر في الأعلى و هذا يعني بأن أي حركة ستقوم بها سوف تسجل داخل Storyboard ، لاحظ أيضا الشريط الزمني على اليمين و فوقه شيء شبه النجمة من خلالها تستطيع أن تضيف فواصل زمنية بين كل حركة وحركة . أيضا لديك شريط التحكم لمشاهدة ماذا فعلت ،(انظر الصورة في الاسفل):



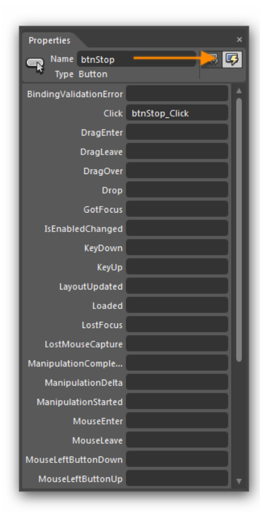
لعمل تأثير الشمس سوف نقوم بتصغير الدائرة أولا و من ثم تكبيرها من جديد ، بعد إضافة Storyboard حدد دائرة الشمس من نافذة OATL ثم قم بسحب الخط الزمني الأصفر حتى الثانية الأولى اضغط على زر النجمة من الأعلى لتضيف نقطة فصل ثم قم بتصغير دائرة الشمس باتجاه الغرب قليلا ، كرر عملية التصغير من جديد إلى الثانية الثالثة بعد ذلك قم بتكبير الدائرة من جديد حتى الثانية السابعة ، (انظر الشكل التالي) :



للأسف لا تستطيع أن تتحكم بجعل هذا animation يتحرك إلا من خلال الشفرة ، لذلك من نافذة projects حدد النافذة MainPage ثم اضغط على السهم بجانبها لترى صفحة الشفرة MainPage.xaml.cs في المشيد الرئيسي أضف الشفرة التالي:

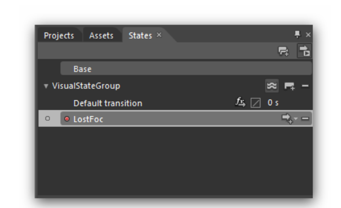
```
1:   Storyboard1.Begin();
```

اضغط على ctrl + shift + B ، لبناء التطبيق ثم انقر على زر F5 للبدء بالعمل و انظر لعملية التحريك كيف تتم ، ما زالت لدينا مشكلة هنا ، أنك لا تستطيع أن تقوم بتفعيل Storyboard بعد توقفه ، يمكنك استخدام Button مثلا و تفعيلها من هناك ، لديك العديد من الطرق فيما يخص Storyboard و هي تتيح لك التحكم بشكل كلي في animation في Silverlight و WPF ، تستطيع كتابة تلك الطرق من خلال الضغط على زر الأحداث الموجود بجانب الخصائص (انظر الصورة على اليسار) ، قم بتحديد الحدث Click للأداة Button و اكتب الطرق بالشكل التالي :

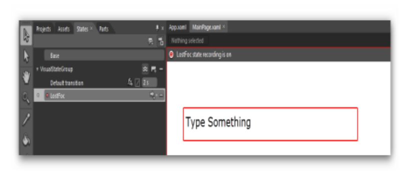



```
1: Storyboard1.Pause();
2: Storyboard1.Stop();
```

هناك طريقة أخرى تختلف تماما عن هذه الطريقة و هي تعتمد على الأحداث بشكل كلي ، مثلا عندما يغادر التركيز صندوق النص يتم إطلاق هذا الحدث ، حيث تعرف هذه الطريقة بـ status ، ارسم TextBox على نفس النموذج في مكان آخر (ستجد أداة TextBox في النافذة Assets) من النافذة status أضف status جديد (الزر الأيمن في الأعلى) .



ستحمل النافذة عنوان جديد تحت مسمى visualStatusGroup ، في هذه الأخيرة انقر على زر إضافة ليتم إضافة visualStatus ، غيّر اسمها مثلا إلى LostFoc (انظر الصورة على اليمين) . لاحظ الآن أن الأمر يعود مجددا فتصبح منطقة التصميم باللون الأحمر استعدادا للتسجيل ، من خلال status لا تحتاج إلى التعامل مع TimeLine بشكل مباشر ، فقط قم بتغير المدة الزمنية إلى مدة معينة و لنقل مثلا ثانيتين ، و ذلك من خلال تغيير Default translation إلى ٢.٠ .



الآن حدد TextBox و توجه إلى قائمة الألوان من شريط الخصائص و حدد BorderBrush من الأعلى ، ثم قم بتغيير لون التصميم إلى ما يحلو لك ، هنا سأجعلها باللون الأحمر (انظر الصورة على اليسار) ، بعد أن تنتهي من تسجيل تغيير اللون ، عد إلى شريط status و غير الحالة إلى base وذلك بالضغط عليها فقط .

الخطوة الأخيرة هي إطلاق هذا الحدث عند الخروج من TextBox مباشرة ، لعمل ذلك انتقل إلى صندوق الأدوات Assets و اختر هذه المرة التصنيف Behaviors ، ثم اختر الأداة GoToStatusAction أو فقط ابحث عنها في الشريط من الأعلى . ☺



قم بسحب الأداة وإلقائها على TextBox في النافذة OATL أو على TextBox في منطقة التصميم ، ستظهر على اليمين خصائص هذه الأداة الجديدة ، حدد من Trigger الخاصية EventName واختر MouseLeave ، و من Common Properties حدد الخاصية Status Name و اختر الـ LostFoc ، خطوة أخيرة قبل أن تقوم بتجربة التطبيق أضف button إلى النموذج، حتى تستطيع أن ترى الحدث LostFoc ، نفذ البرنامج الآن و انظر إلى النتيجة .

أحد أهم المميزات في تطبيقات Silverlight أنه يمكنك تعديل أي شيء فيها تقريبا، لنأخذ على سبيل المثال صندوق النص ، إذا أردت أن تعرف كيف تم تطويره بالكامل ، فكل ما عليك عمله هو النقر على أي TextBox و اختر Edit Temples و من ثم اضغط على OK و استمتع بمذاق شفرة XAML شهي ☺ (انظر الصورة في الأسفل). يمكنك أيضا تعديل style هذه الأدوات بشكل عادي جدا كما شاهدت في هذا الفصل .

على الرغم أننا لم ننجز شيئا عمليا حتى الآن ما زلنا في الأساسيات و ما زال الكثير في Blend خفيا و لكن تعرفنا على مكامن قوة Blend ، التعامل مع animation كان سابقا معجزة علمية بحد ذاتها ، رأينا في هذا الفصل كيف يتم التعامل معها بطريقتين سواء status أو Storyboard و رأينا كيف يتم التعامل مع الألوان و التصاميم ، أيضا تطرقنا لأهم النوافذ في Blend و طرق التعامل معها . أخيرا رأينا كيف يمكنك الحصول على شفرة أي أداة في Silverlight و كيف يمكنك الاطلاع عليها .



الفصل الثاني :

أدوات Silverlight القياسية

في هذا الفصل سنتطرق لأدوات Silverlight الأساسية مثل Button و TextBox و ListBox و غيرها ، سنتعرف على الفروقات بين أدوات Silverlight و windows form ، سنرى أكثر الخصائص استخداما لهذه الأدوات و أكثر الأحداث التي يتم التعامل معها خصوصا عند الحديث عن مطور التطبيقات Windows form ستجد في هذا الفصل العديد من الملاحظات التي تبين الفروقات بينهما أثناء شرح كل أداة .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- استخدام أدوات Silverlight القياسية .
- التعامل مع data binding .
- التحقق من البيانات و Data validation .

استخدام أدوات Silverlight القياسية :

أولا الخصائص المشتركة بين الأدوات :

- الخاصية FlowDirection : و تستخدم لتحديد اتجاه النص سواء من اليمين لليسار لدعم اللغات مثل العربية أو من اليسار لليمين للإنجليزية مثلا ، هناك طريقتين للتحكم بهذه الخاصية سواء عن طريق نافذة الخصائص أو من خلال الشفرة :

```
1: TextBox1.FlowDirection=
2: System.Windows.FlowDirection.RightToLeft;
```

- الخاصية Text : و تستخدم لتعيين النص أو الحصول على قيمة من TextBox مثلا، يمكنك تعيين النص من نافذة الخصائص أو من خلال الشفرة ، و هي تستقبل و تعيد متغير من النوع string .
- الخاصية IsEnable : و هيا توازي الخاصية Enable في تطبيقات windows form و تعيد و تستقبل متغير من النوع Boolean ، حيث تستخدم هذه الخاصية لتمكين المستخدم من التفاعل مع TextBox أو عدم التفاعل معه .



- الخاصية MaxLength : و تستخدم لتعيين أقصى طول للحروف في TextBox و تستقبل متغير من النوع Int يحدد عدد الحروف .
- الخاصية Viability : و تستخدم لعرض أو اخفاء الأدوات أثناء التشغيل ، وتأخذ قيمة من النوع Viability و هي توازي الخاصية visible في تطبيقات windows form ، أخيرا يمكنك تعديل هذه الخاصية من خلال الشفرة بالطريقة التالية :

```
1: TextBox1.Visibility = System.Windows.Visibility.Visible;
```

- الخاصية isReadOnly : و تستخدم لجعل بعض الأدوات للقراءة فقط ، أرجو أن تلاحظ أن الأمر مختلف عند استخدام isEnable ، على عكس IsEnable يستطيع المستخدم التعامل مع الأداة و لكن لا يستطيع الكتابة فيها مثلا بإمكانه نسخ محتويات النص و ليس لصق النص . و هي تتعامل مع قيم من النوع Boolean .
- الخاصيتين VerticalScrollbar و HorizontalScorllbar :

كلا الخاصيتين تستخدم لعرض شريط التمرير ، يمكن تعيين أربعة قيم لهذه الخاصية إما hidden أي مخفية أو visible أو Auto و هنا يتم تعيين عملية الإظهار بشكل تلقائي عند كتابة النص ، و أخيرا القيمة الرابعة و هي Disabled و من خلالها لا يستطيع أن يرى المستخدم أي من هذه الأشرطة .





هناك خصائص أخرى تتعلق بالتصميم مثل style و Brushes color و لون الخلفية و الخطوط و قد تطرقنا لهذه الأمور جميعها و رأينا كيف يتم التعامل معها من خلال Blend ، نصيحة إنسانية إذا أردت أن تتعامل مع التصميم لا تستخدم Visual studio فستجد نفسك في متاهة لا يمكن أن تخرج منها إطلاقا دون التعامل مع XAML .

ثانيا الأحداث المشتركة :

- الحدث GetFocus و LostFocus : الحدث الأول عكس الحدث الثاني فالأول يتم إطلاقه عندما تحصل الأداة على التركيز ، أما الأخر فيتم إطلاقه عندما تخسر الأداة التركيز ، فكر في TextBox التي تتعامل معها twitter مثلا عند عملية البحث ستجد كلمة search عندما تضع مؤشر الفأرة داخل TextBox ستلاحظ اختفاء كلمة search و هنا أطلق الحدث GetFocus و عندما تخرج من TextBox دون كتابة أي كلمات تعود كلمة search مجددا و هنا تم إطلاق الحدث LostFocus.



- الحدثين KeyDown و keyUp : يقع الحدث الأول عندما يقوم المستخدم بالنقر على زر من لوحة المفاتيح و قبل أن يرفع أصبعه من الزر ، و أما الحدث الثاني فيقع عندما يرفع المستخدم يده من لوحة المفاتيح ، مثلا يمكن أن تقوم بعمل أمر ما عندما يقوم المستخدم بالنقر على زر enter من لوحة المفاتيح ، الشفرة:

```
1:  private void TextBox2_KeyDown(object sender,
2:  KeyEventArgs e)
3:  {
4:    if (e.Key == Key.Enter)
5:      {
6:       MessageBox.Show(" hi ");
7:      }
8:  }
```

لاحظ في الشفرة التي في الأعلى أني أستخدم المتغير e لتعقب حركة المستخدم ، سيقع هذا الحدث عند عملية النقر على زر Enter و قبل رفع يديه من لوحة المفاتيح ، حاول أن تقوم بنفس الأمر و لكن بعد عملية النقر !

- لدينا الحدثين MouseEnter و MouseLeave : يقع الحدث الأول عندما يكون مؤشر الفأرة فوق الأداة مباشرة ، أما الحدث الثاني فيقع عندما يخرج مؤشر الفأرة من نطاق الأداة .



- الأحداث MouseDown and MouseUp (left and right): و هي تقع عندما يقوم المستخدم باستخدام زر الفأرة الأيمن أو الأيسر ، الحدث Down يقع عندما يتم النقر بزر الفأرة و قبل رفع الأصبع عن الزر أما الحدث الأخر فيقع عندما يقوم المستخدم بإطلاق زر الفأرة و هو up ، لاحظ هنا لدي أربعة أحداث اثنين down للزر الأيمن و الزر الأيسر و اثنين Up للزرين الأيمن و الأيسر أيضا.

- الأدوات TextBox و TextBlock و label :

جميع هذه الأدوات تتعامل مع النصوص سواء في السماح للمستخدم بإدخال النصوص أو عرض النصوص عليه ، حسنا لنبدأ أولا بـ TextBox ، حيث تسمح هذه الأداة بإدخال النصوص من قبل المستخدم مثلا سؤال المستخدم عن عمره أو اسمه .. الخ .

- الحدث TextChange : و يقع هذا الحدث مباشرة عند تغيير محتوى TextBox و هو خاص بهذه الأداة .

- الأداة label :

تستخدم الأداة label لعرض نصوص توضيحية على المستخدم مثل تعليمات أو شيء من هذا القبيل ، لا يوجد الكثير للحديث عنه في هذه الأداة فهي بسيطة جدا و أكثر الأدوات شعبية في جميع التطبيقات تقريبا ، و لكنها في Silverlight مختلفة جذريا ☺ .



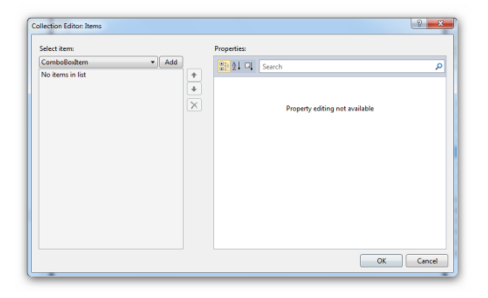
الخصائص :

- الخاصية Content : على عكس label العادي في تطبيقات asp.net و تطبيقات windows form ليس لدينا هنا خاصية Text لهذه الأداة ، عوضا عن ذلك لدينا خاصية اسمها Content وهي خاصية يمكنها أن تحتوي على أي شيء ! نعم أي شيء حتى أدوات أخرى مثل Button و صناديق نصوص أخرى ! ، شفرة XAML التالية ستوضح لك كيف بإمكانك أن تضع button داخل أداة label ، الشفرة :

```
1: <sdk:Label Height="89"
2: HorizontalAlignment="Left"
3: Margin="78,140,0,0" Name="label1"
4: VerticalAlignment="Top" Width="250">
5:   <Button Height="64" Width="136">
6:   </Button>
7: </sdk:Label>
```

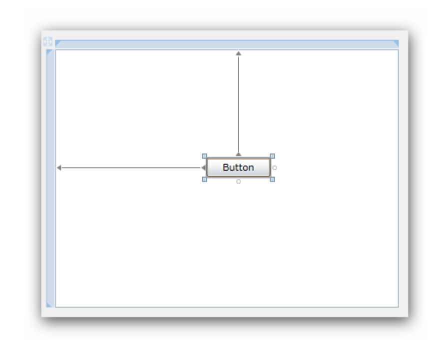
- الأداة TextBlock : و هي الأداة الموازية لأداة label في تطبيقات windows form و تطبيقات asp.net ، أحداث هذه الأداة متطابقة كليا مع القائمة في الأعلى ، يمكنك استخدام الخاصية Text للحصول على قيمة ما .

- الأداتين CheckBox و radioButtun : الأولى تستخدم للحصول على عدة خيارات و الأخيرة تستخدم للحصول على خيار واحد من مجموعة خيارات ، لا يوجد أي جديد في هذه الأدوات بالنسبة لـ Silverlight أحداثها أيضا هي نفس الأحداث السابقة تقريبا اختلاف بسيط فقط في التسمية للحصول على قيمة هذه الأداتين بدلا من Check أصبحت IsChecked .
- الأداتين ListBox و Combbox : حسنا هنا لدينا اختلاف جذري في كلتا الأداتين عند مقارنتهما بـ windows form ، أولا ListBox أصبح أقرب إلى FlowLayoutPanle و في نفس الوقت هي ListBox ، بمعنى أنه يمكن أن تحتوي هذه الأداة على أدوات أخرى و في نفس الوقت تحتوي على قيم نصية ، أما Combobox فقد أصبح بإمكانك أن تجعلها تحتوي أيضا على العديد من الأدوات كما تم اضافة حدثين جديدين للـ Combobox و هما DropDwonClose و يطلق عند إغلاق القائمة و أما الحدث الثاني DropDownOpen فيطلق عند فتح القائمة مباشرة ، يمكنك إضافة عناصر إلى هاتين الأداتين عن طريق الخاصية items ، و أعتقد طريقة الإضافة بسيطة جدا ، بالنسبة للإضافة من خلال الكود هي نفس الشفرة في تطبيقات windows form أو تطبيقات asp.net لم يتغير شيء ، مثال :




```
1: comboBox1.Items.Add(" test");
```

- الأداتان Grid و DataGrid : حسنا هاتان الأداتان من أكثر الأدوات التي ستتعامل معها ، خصوصا الأداة الثانية عند الحديث عن قواعد البيانات ، بنسبة للأداة الأولى و هي Grid فهي أداة يمكنك من خلالها أن تضيف لها مجموعة من الأدوات ، وهنا يجب الحديث عن جزء أخر من عملية التصميم ، لنفترض أن لديك Button على الشاشة هناك شيء يسمى بمقابض التحكم و هيا أربعة مقابض لكل أداة (الصورة على اليمين) ، أحد عيوب Silverlight أنه ليس بالضرورة ما تراه أثناء التصميم هو ما ستراه أثناء التنفيذ ، اسحب الـ Button إلى الأسفل ثم إلى أقصى اليمين مع ملاحظة أن تجعل المقابض المفعلة المقبض بالجهة اليسرى و المقبض في الجهة العليا ، شغل التطبيق الآن ، انظر إلى النتيجة ، لم ينزل الزر إلى الأسفل في أقصى اليمين كما من المفترض أن يكون ، أغلق المتصفح و فعل هذه المرة المقبض في الجهة اليمنى و المقبض في الجهة السفلى ، مع إلغاء المقابض العليا و اليسرى . أعد تشغيل التطبيق و انظر إلى الكارثة التي أمامك !!



هناك حل لهذه المشكلة و هي تحديد حجم النموذج بحجم المتصفح عوضا عن جعل حجم النموذج تلقائيا ، لكن في هذا الحال ستعرض مشروعك بالكامل في مواجهة مع سلاح الدمار الشامل في عالم تصميم صفحات الويب و هو طريقة العرض لدى المستخدم أو Resolutions ، إذا كان حجم العرض في جهاز المستخدم يختلف عن

طريقة العرض في جهازك ربما لن تجد أزرار أصلا و لن يجد أصلا صفحة Silverlight لكي يتعامل معها المستخدم.



إذا من غير المجدي التفكير بتكبير النموذج ، هذه المشكلة و هي عدم تحجيم النموذج ستولد كارثة أثناء تصميم النماذج الكبيرة نوعا ما (انظر الصورة على اليسار) ، ماذا تلاحظ ؟ نعم لقد أصبحت الأدوات متداخلة أثناء التصميم و لكن الأمر ليس كما تراه أثناء التنفيذ ، و هنا يأتي دور الأداة Grid فبدلا من تحديد مقابض المحاذاة لكل عنصر على حده ، يمكنك فقط أن تضيف مجموعة من الأدوات إلى Grid معين و من ثم تهتم فقط في مقابض الـ Grid ، هل تستطيع أن تحزر كم عدد الـ Grid الموجود في الصورة على اليسار ؟! ☺ .

الأداة DataGrid : رغم التشابه الكبير في طريقة العمل لهذه الأداة مع شقيقتها في Windows form إلا أنها مختلفة نوعا ما في طريقة التعامل مع البيانات ، في البداية ابدأ مشروع جديد من نوع Silveright و أضف كلاس جديد و ليكن بعنوان MyFavoWebSite (لاحظ أنك ستضيفه الى تطبيق silverlight وليس تطبيق Web)، ثم أضف هذه الخصائص لهذا الكلاس :

```
1:  public class MyFavoWebSite
2:  {
3:      public string name { get; set; }
4:      public string link { get; set; }
5:      public bool isOnline { get; set; }
6:  }
```



في الصفحة الرئيسية لمشروع silverlight و داخل مشيد الصفحة سنقوم بإنشاء List من هذا الكلاس ، مثلا بالمعلومات التالية :

```
1:  List<MyFavoWebSite>
2:        Li = new List<MyFavoWebSite>();
3: MyFavoWebSite Number_1 = new MyFavoWebSite();
4:    Number_1.isOnline = false;
5:    Number_1.link = "www.google.com";
6:    Number_1.name = " google ";
7:
8:    MyFavoWebSite Number_2 = new MyFavoWebSite();
9:    Number_2.isOnline = true;
10:   Number_2.name = "Silverlight";
11:   Number_2.link = "www.silverlight.net";
12:
13:   Li.Add(Number_1);
14:   Li.Add(Number_2);
15:   dataGrid1.ItemsSource = Li;
```

لا تنسى أن تقوم بإضافة datagrid إلى النموذج ، شغل التطبيق الآن و انظر إلى النتيجة ، أعرف لم يحدث أي شيء حتى الآن و لكن إليك قليلا من السحر ، أظف هذا السطر قبل تحديد الخاصية ItemSource بالشكل التالي :

```
1:  dataGrid1.AutoGenerateColumns = true;
```

قم بتشغيل التطبيق الآن ، هل تغير الأمر ، حاول تحريك الأعمدة و تعديل المحتوى ، كل شيء أصبح معد كليا دون أن تكتب شفرة واحدة لتحديد طريقة العرض ، هنالك المزيد من هذه المفاجئات (بس صبرك عليا شوية ☺) ، أكثر الأحداث تعاملا في هذه الأداة هي selectedchange و هي عنصر من نوع object ، مثلا للعودة بالقيمة التي قمت بتحديدها في مثالنا هنا ، باستخدام سي شارب :

```
1: private void dataGrid1_SelectionChanged(object sender
2:             , SelectionChangedEventArgs e)
3:  {
4:    MessageBox.Show(((MyFavoWebSite)dataGrid1.
5: SelectedItem).name);
6:  }
```

جميل صحيح ☺ ، بما أن القيمة المعادة عبارة عن Object فيمكنك تحويلها كما تشاء ، و بما أن العناصر في داخل Datagrid من النوع MyFavoWebSite لذلك جرى تحويلها لهذا النوع و من ثم يمكنك الوصول لخصائص هذا التصنيف بشكل طبيعي .

كما يمكنك تعديل عناصر هذه الأداة بشكل يدوي عن طريق الخاصية Collection ، يمكنك تفعيل و إلغاء تفعيل الاختيار المتعدد عن طريق الخاصية SelectionMode و التي تستقبل إما Extended و هيا الاختيار المتعدد أو Single و هو الاختيار المفرد .

أخيرا يمكن الحصول على بيانات الخلية المحددة دون كتابة شفرة سي شارب واحدة و هو ما سنراه في الفقرة التالية عند الحديث عن Data Binding .



ضم البيانات بين الأدوات أو data Binding :

رأينا في المثال الذي في الأعلى كيف يتم الحصول على البيانات من DataGrid باستخدام سي شارب ، هناك ميزة عظيمة أخرى في XAML و هي ما تعرف بـ data Binding ، حيث تمكنك هذه الأخيرة من الحصول على بيانات الأدوات أثناء التصميم دون كتابة شفرة سي شارب واحدة .

فكر منطقيا لماذا علينا استخدام شفرة سي شارب إذا كنا نرغب في نقل محتويات العنصر المحدد في datagrid إلى TextBox مثلا ، فكر كم حجم الشفرة التي ستكتبها لعمل ذلك ، هنا تأتي فكرة data binding ، لتحل هذه المشكلة عوضا عن استخدام سي شارب أو VB يمكن استخدام XAML لربط الأدوات ببعض و جعل عملية تناقل المعلومات أسهل بكثير و أن تتولى Silverlight السيناريو السابق من الإفراغ و التعبئة من جديد ، إليك كيف يتم الأمر :



لقد حاولت أن أبسط الفكرة قدر المستطاع ، وفقا للصورة السابقة لنفترض أن لدينا السيناريو كما في الصورة التي على اليسار ، ما أريد عمله هنا تفعيل أو عدم تفعيل صندوق النص تبعا لتحديد الأداة RadioButton Enable ، اذا لنقم بتفصيل العمل قليلا ، كما في الأعلى :



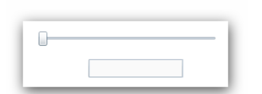
- الأداة التي ستتأثر أو تأخذ البيانات هي TextBox .
- سيتم تنفيذ البيانات في الخاصية IsEnable .
- اسم الأداة التي سنأخذ منها البيانات RadioButton وتحمل الاسم radEnale .
- اسم الخاصية التي سنحصل منها على البيانات هي IsChecked .

كم تتوقع حجم الشفرة لعمل هذا الأمر ☺ ، انظر الى السطر في الأسفل :

```
1: <TextBox IsEnabled="{Binding
2: ElementName=radEnable , Path=IsChecked}" />
```

فقط !

مثال أخر ليكن لدينا أداة Slider و أريد أن أقوم بتغيير قيمة TextBox هذه المرة بقيمة محددة في Slider، (انظر الصورة على اليسار)، لنرتب ماذا نريد أن نفعل :





- الأداة التي ستتأثر هي TextBox .
- سيتم تنفيذ البيانات على الخاصية Text .
- اسم الأداة التي سنأخذ منها البيانات Silder1 .
- اسم الخاصية value .

فقط أضف السطر التالي إلى سطر صندوق النص في XAML :

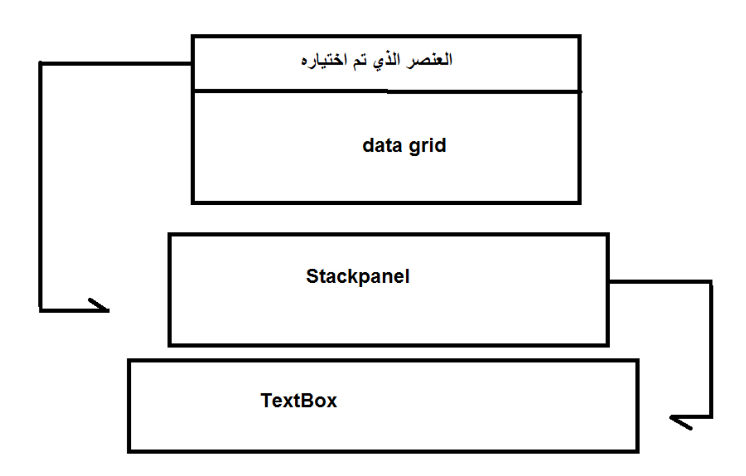
```
1: Text="{Binding ElementName=slider1,Path=Value}"
```

الآن لنعد إلى مثالنا السابق و المتعلق بـ datagrid ، هنا لدينا مشكلة ، فعدد القيم التي يتم اختيارها هذه المرة ليست واحدة ، ما أقصده هنا أن كل صف في datagrid به ثلاثة قيم ، ما أريد الحصول عليه هنا هو الخاصية name ، لعمل مثل هذا الأمر نحن بحاجة إلى طبقة وسيطة بحيث تستضيف هذه الطبقة العنصر الذي ثم تحديده من الـ datagrid ، ثم من خلال هذا العنصر أستطيع أن أصل إلى الخاصية name الموجودة فيه ، هذا العنصر الوسيط عبارة عن Stackpanel .

StackPanel هي أداة شبيهة جدا لـ grid إلا أنها تفرض عليك أسلوب ترتيب معين للأدوات التي بداخلها، هي مفيدة جدا عندما تريد التأكد كليا من أن مجموعة من الأدوات لها ترتيب معين متناسقة في الطول و الارتفاع .



في المثال السابق دع datagrid و شأنها ، و لكن أضف StackPanel إلى النموذج و ضع بداخلها TextBox ، (انظر الصورة لشرح طريقة العمل) :



لأخذ البيانات من datagrid سنحتاج للخاصية SelectedItme ، أما الخاصية المستهدفة في Stackpanel هي dataContext ، الشفرة :

```
1:  <StackPanel DataContext="{Binding
2:                ElementName=dataGrid1,  Path=SelectedItem}"/>
```

بعد الحصول على العنصر الذي تم تحديده سننقله إلى Textbox و بما أنه Object من النوع MyFavoWebSite بالتالي سيحتوي على الخاصية name ، إذا الخاصية التي سنحصل عليها من البيانات هي name و الخاصية التي ستتأثر هي Text ، هنا لسنا بحاجة لتحديد المصدر لأن المصدر تلقائيا سوف يكون StackPanel ، الشفرة :

```
1:  <TextBox Text="{Binding Path=name}" />
```



في نهاية الأمر تعتمد تطبيقات Silverlight بشكل كبير جدا على مفهوم data binding لم يعد يقوم بضم البيانات بين الأدوات من خلال سي شارب أو VB ، لا أبالغ إن أخبرتك أنه من النادر جدا أن ترى مثل هذا الأمر يحدث في تطبيقات Silverlight الجيدة على الأقل ، لذلك من المهم جدا أن تفهم هذه النقطة ، إذا لم أفلح في توضيح هذه النقطة راجع صديقنا الصدوق Google فستجد كم هائل من المقالات حول هذه النقطة .

التحقق من البيانات و Data validation

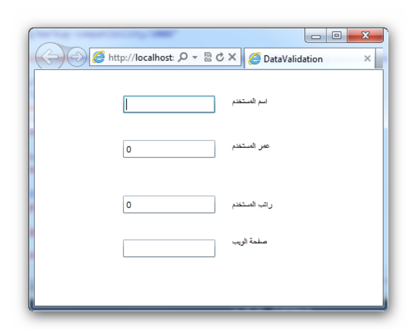
الجزء الأخير من هذا الفصل سنتحدث فيه عن أحد الأمور الأساسية في عالم Silverlight و هي التحقق من صحة البيانات ، رأينا في الفصل السابق كيف تتم عملية ضم البيانات و Data binding و الآن سنستخدم هذه التقنية ، جعل Silverlight تقوم بشكل شيء عنا .

لنفترض أن لدينا التصنيف التالي و الذي يمثل موظف :

```
1:  public class Emp
2:      {
3:    public string name { get; set; }
4:    public int age { get; set; }
5:    public string web { get; set; }
4:    public double salary { get; set; }
6:      }
```

في الخطوة التالية سنقوم بإنشاء كائن من هذا التصنيف و سوف نقوم بعملية Binding لهذا الكائن في النموذج و ذلك في المشيد الرئيسي للصفحة ، الشفرة :

```
1:  public MainPage()
2:  {
3:      InitializeComponent();
4:      Emp Omar = new Emp();
5:      this.DataContext = Omar;
6:  }
```



ثم لدينا تصميم بسيط لهذه الصفحة فيه أربعة عناصر TextBox، كل الصناديق تم عمل Binding لها مع خصائص الكائن Omar ، انظر إلى شفرة XAML و الصورة على اليمين للنموذج :

```
1:  <TextBox Text="{Binding name}"/>
2:  <TextBox Text="{Binding age}"/>
3:  <TextBox Text="{Binding salary}"/>
4:  <TextBox Text="{Binding web}"/>
```

إذا قمت بتشغيل التطبيق الأن ستلاحظ ظهور الرقم ٠ عند حقلي العمر و الراتب .

ما أريد فعله هنا هو التحقق من مدخلات المستخدم في حقلي العمر و الاسم بحيث لا يبدأ الاسم بالحرف D مثلا و الراتب بحيث لا يتجاوز ٥٠٠٠ ريال .



لنبدأ أولا بالعمر و الراتب ، من خلال سي شارب أحد أكثر الطرق المتبعة للتأكد من أن البيانات المدخلة هي من النوع Integer هي تحويل القيمة إلى integer فإن تمت بشكل صحيح لنكمل في الطريق و إن لم تتم العملية بشكل صحيح نخبر المستخدم بضرورة تصحيح الأمر . ربما تكتب شيء من هذا القبيل مثلا :

```
1: public bool CheckNumberAsInteger(string number)
2:           {
3:      Try
4:           {
5:             int.Parse(number);
6:             return true;
7:           }
8:      catch { return false; }
9: }
```

لكن أنت الآن في عالم Silverlight حيث كل شيء أبسط مما تتخيل ☺ ، جرب فقط اضافة خاصية ValidatesOnExceptions إلى الوسم Data Binding في XAML ، بهذا الشكل إلى صناديق العمر و الراتب :

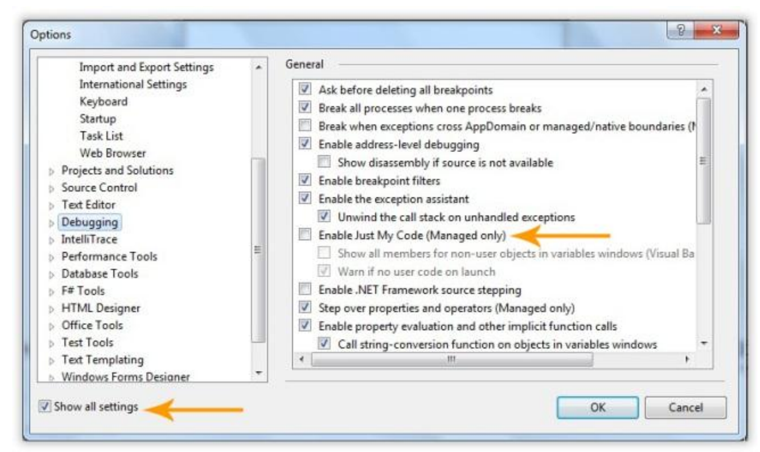
```
1: Text="{Binding
2: age,Mode=TwoWay,ValidatesOnExceptions=True}"
```

فقط ! شغل التطبيق الآن و انظر إلى النتيجة وادعو لي ☺ .

حسنا ما الفرق بين ترك مهمة التحقق إلى Silverlight و بين أن تكون أنت المسؤول عنها ، أستاذي العزيز بعيدا عن الشفرة مهما بلغت درجة لمستك الفنية في إبلاغ المستخدم بهذا الخطأ لن تكون أجمل من طريقة Silverlight . بالإضافة طبعا إلى سهولة الشفرة كليا مقارنة مع شفرة سي شارب .

حسنا لنكمل، بالنسبة لحقل الراتب ما زال لدينا مشكلة فأنا هنا أريد أن أجعل الراتب لا يزيد عن ٥٠٠٠ ريال ، الجزء الأول من عملية التحقق و المتعلق بأن القيمة لابد أن تكون رقما تم حلها و لكن بقي الجزء الثاني.

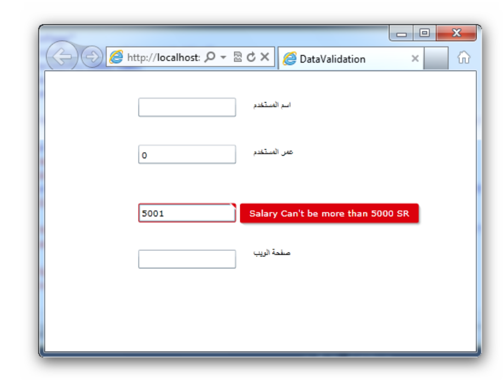
قبل أن نكمل في الجزء الثاني هناك خاصية يجب ايقافها في VS ، و هي خاصية enable just my code و الهدف من إيقاف هذه الخاصية هو جعل Silverlight تعالج الخطأ و ليس VS ، لإيقاف هذه الخاصية توجه إلى Tools > Option > Debugging > Enable Jest My Code و قم بإلغاء تفعيلها إذا لم تظهر لك جميع الخيارات قم بتفعيل الخاصية Show all setting (انظر الصورة في الأسفل).





سنقوم بتعديل شفرة سي شارب السابقة الخاصة بالكائن Emp ، بالشكل التالي :

```
1:  public class Emp
2:          {
3:    public string name { get; set; }
4:    public int age { get; set; }
5:    public string web { get; set; }
6:
7:    double _salary;
8:    public double salary {
9:    get { return _salary; }
10:   set
11:     {
12:       if (value > 5000)
13:         {
14: throw new Exception("Salary Can't be more than 5000 SR");
15:          }
16: _salary = value;
17:     }
18:   }
19: }
```



لاحظ الخاصية لم أقم بتغيير أي شيء في الواقع سوى طريقة التعامل مع الخاصية salary لاحظ كيف تتم معالجة قيمة salary و كيف يتم التأكد من أنها أكبر من 5000 إذا تحقق الشرط أقوم برمي Exception ومعه الخطأ ، أعد تشغيل التطبيق الآن و أدخل قيمة أكبر من ٥٠٠٠ و انظر ماذا يحدث ؟ بالنسبة لخاصية الاسم ، يمكنك عملها بهذه الطريقة :

```
1:      string _name;
2:              public string name
3:          {
4:      get { return _name; }
5:      set
6:          {
7:              if (value.StartsWith("D"))
8:                  {
9:  throw new Exception("name can't start with Letter D ");
10:                 }
11:             _name = value;
12:         }
13:         }
```

مازال في جعبتي المزيد حول data validation ، لنكمل النقطة التالية :



ماذا إذا أردت أن تتحكم بعملية التحقق ، مثلا تريد أن تظهر عنصر TextBox الذي يحتوي على الخطأ بخلفية صفراء ، هناك طريقتين إما تعديل Stats الخاص بعملية التحقق من Blend كما رأينا ، أو باستخدام سي شارب ، لنجرب الأمر على TextBox الخاص بالراتب ، في البداية علينا إضافة وسم و جعل قيمته true إلى TextBox، هذا الوسم يخبر Silverlight بأنك تريد التحكم بعملية التحقق

```
1: NotifyOnValidationError=True
```

طبعا يتم إضافة هذا الوسم بجانب الوسوم الأخرى في الخاصية Binding لأداة TextBox ، بما أنك تريد التحكم بعملية التحقق ، تذكر شيء مهم دائما يمكنك التحقق من أداة معينة من خارج أداة أكبر منها بمعنى يمكنك التحكم بـ Buttun موجود داخل StackPanel من StackPanel ، و يمكنك التحكم بكل شيء على النموذج من خلال Grid طبعا التي تحمل الاسم LayoutRoot ، من شريط الخصائص قم بالضغط على الحدث BindingValidationError ينطلق عند حدوث أي أخطاء في كل الأدوات الموجودة داخل هذه الـ grid . اكتب الشفرة التالية بداخله ، الشفرة :

```
1: if (e.Action == ValidationErrorEventAction.Added)
2:             {
3: (e.OriginalSource as Control).Background =
4:     new SolidColorBrush(Colors.Yellow);
5: ToolTipService.SetToolTip((e.OriginalSource as TextBox),
6: e.Error.Exception.Message);
7: }
8: if (e.Action == ValidationErrorEventAction.Removed)
```

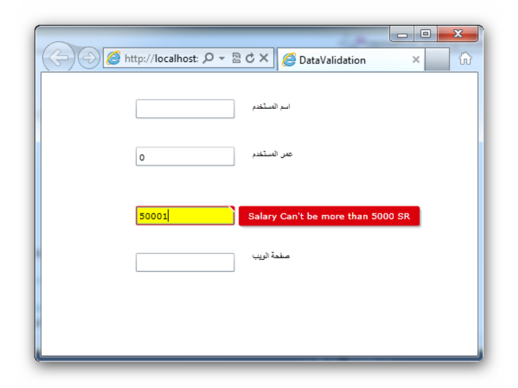



```
9:  {
10: (e.OriginalSource as Control).Background =
11: new SolidColorBrush(Colors.White);
12: ToolTipService.SetToolTip((e.OriginalSource as
13: TextBox),null);
14: }
```

إذا عدت إلى شفرة XAML ستجد أن VS قد أضاف وسم جديد للحدث ، الشفرة:

```
1: <Grid x:Name="LayoutRoot" Background="White"
2: BindingValidationError="LayoutRoot_BindingValidationError">
```

عند وجود خطأ هناك طريقتين إما وجود الخطأ و هو يتمثل بـ Add أو انتهى الخطأ وهو يتمثل بـ Removed ، ما أقوم بعمله هنا هو أن المتغير e لديه خاصية اسمها OriginalSource وهو عبارة عن كائن يمثل أي شيء يطلق هذا الخطأ و بما أني أريد معالجة الأخطاء التي تنطلق من أدوات TextBox فالأخيرة عبارة عن Contorls فأقوم بتغيير لون الخلفية حيث تأخذ الأخيرة SolidColorBrush



كما رأينا في Blend، أخيرا أضف ToolTipService ومرر لها الخطأ الذي وقع . في Removed يحدث العكس أعيد TextBox إلى اللون الأبيض و أجعل قيمة ToolTip فارغة ، (انظر الصورة على اليمين) ، هل انتهينا من Data validation ، كلا لم ننتهي بعد (صبرك علينا يا حلو ☺) ، لنكمل .



بقي لدينا نقطة أخيرة عند الحديث عن التحقق من البيانات و هي ملخص الأخطاء ، باستخدام أدوات تسمى ValidationSummary يمكنك عرض ملخص لكل الأخطاء التي وقع بها المستخدم أثناء تعبئة النموذج ، في البداية علينا إضافة References إلى مشروع Silverlight حدد المشروع ثم انقر بزر الفأرة الأيمن و اختر Add References ستظهر قائمة References ابحث عن System.Windows.Controls.Data.Input ، الآن علينا إضافة هذا المرجع إلى الصفحة التي نعمل عليها أضف هذ المرجع إلى الشاشة بالشكل التالي :

```
xmlns:ds=""
mc:Ignorab                                                          com/winfx/2
d:DesignHe
<Grid x:Na                                                          ot_BindingV
    <TextB                                                          ight="23" H
    <TextB                                                          ht="23" Hor
    <TextB                                                          tifyOnValid

    System.Windows.Automation (System.Windows)
    System.Windows.Automation.Peers (System.Windows)
    System.Windows.Automation.Peers (System.Windows.Controls.Data.Input)
    System.Windows.Automation.Text (System.Windows)
    System.Windows.Browser (System.Windows.Browser)
    System.Windows.Controls (System.Windows)
    System.Windows.Controls (System.Windows.Controls.Data.Input)
    System.Windows.Controls.Primitives (System.Windows)
    System.Windows.Data (System.Windows)
```

أنزل إلى أسفل شفرة XAML و قبل أن تغلق Grid فقط أضف أداة ValidationSummary :

```
1: <ds:ValidationSummary Grid.Row="4" />
```



لاحظ أن grid.row تمثل عدد الأخطاء التي يمكن أن تعرضها على المستخدم في المرة الواحدة ، عد إلى شاشة التصميم و انظر إلى أداة ملخص التحقق ، حاول تصغير الأداة حتى ترى كل الأدوات من تحتها ، هل تعلم ماذا تبقى ، فقط قم بتشغيل التطبيق بالضغط على F5 .

http://localhost:60399/DataValidationTestPage.aspx
DataValidation

اسم المستخدم D

sdfsa Input is not in a correct format.

راتب المستخدم 50112

صفحة الويب

3 Errors
name name can't start with Letter D
age Input is not in a correct format.
salary Salary Can't be more than 5000 SR

٢

الجزء الثاني: في أعماق Silverlight

في هذا الجزء ستتعرف على العديد من الأمور الأكثر تطويرا في Silverlight مثل التخزين على جهاز العميل و قواعد البيانات من خلال Silverlight و خدمات الويب و الوسائط المتعددة ، سيشكل هذا الجزء نقطة تحول في طريقة تعاملك مع تطبيقات Silverlight لذلك كن مستعدا جيدا ، أرجو أن تلاحظ أني لن أتطرق مثلا إلى ADO.net أو كيف يتم قراءة ملفات XML و ما شابه ذلك فهذا الكتاب ليس موجه لتعلم سي شارب أو ADO.net لذلك لابد أن تمتلك هذه الخبرات سابقا .

بعد الانتهاء من هذا الجزء يجب أن يضاف إلى معرفتك التالي :

- تخزين البيانات على جهاز المستخدم .
- التعامل مع خدمات الويب .
- التعامل مع قواعد البيانات .
- التعامل مع الوسائط المتعددة .



الفصل الثالث :

تخزين البيانات على جهاز المستخدم

إذا أمعنت النظر في الطريقة التي تستخدم لتسجيل الدخول في كل المواقع تقريبا ، أنت تقوم بتسجيل الدخول مرة واحدة على جهازك و في المرة القادمة عندما تدخل على الموقع يتم تسجيل دخولك تلقائيا ، ما فعله الموقع هنا أنه قام بتخزين ما يسمى بـ Cookies على جهاز الزائر ، للأسف لا تدعم 4 Silverlight بشكل مباشر عملية الـ cookies عوضا عن ذلك يتم استخدام Cookies الخاص بـ asp.net ، ستتعرف في هذا الفصل كيف تقوم بإنشاء Cookies خاص بك و كيف تقوم بالقراءة و الكتابة داخل هذا الملف .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- التعرف على IsolatedStorageFile .
- الكتابة و القراءة باستخدام IsolatedStorageFile .



التعرف على IsolatedStorageFile

حسنا ما هو IsolatedStorageFile ، باختصار شديد هو طريقة أو آلية تمكنك من تخزين ملفات على جهاز المستخدم ، سواء كانت هذه الملفات Text أو XML أو أي نوع ملفات خاص بك ، لديه أسلوب خاص بالقراءة و الكتابة كما سنرى لاحقا في هذا الفصل .

هناك مشكلتين عند التعامل مع هذ النوع من الملفات ، أولا لا يسمح لك بكتابة ملفات أكبر من ٢ ميجا ، أيضا تذكر دائما أن أماكن تخزين هذه الملفات سهل جدا العثور عليها لذلك عند تخزين كلمات المرور عليك دائما أن تتذكر بضرورة تشفيرها ، أيضا إذا كانت المساحة التي ستكتب بها الملف غير كافية سيطلق Silverlight خطأ بشكل مباشر ، أخيرا يمكن للمستخدم أن يمنع كتابة أية ملفات على جهازه بكل سهولة أو حتى أن يسمح لك بالوصول إلى القرص الصلب في جهازه

لدى IsolatedStorageFile العديد من الطرق أهما ما يلي :

اسم الطريقة	العمل
CreateDirectory	إنشاء مجلد
CreateFile	إنشاء ملف
DeleteDirectory	حذف مجلد
DeleteFile	حذف ملف
DirectoryExists	التأكد من وجود مجلد
FileExists	التأكد من وجود ملف

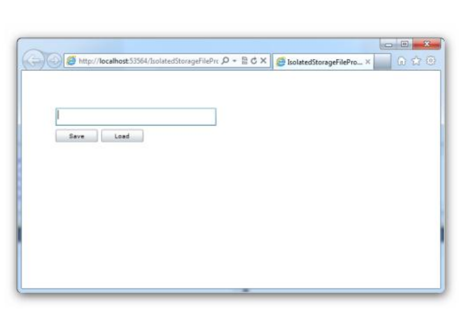


MoveDirectory	نسخ أو نقل مجلد
MoveFile	نسخ أو نقل ملف



هناك أيضا العديد من الطرق الأخرى ، يمكنك مراجعة ملفات MSDN للحصول على معلومات أكثر .

الكتابة و القراءة باستخدام IsolatedStorageFile .



لنفترض أن لدينا TextBox و أداتين Button الأول لحفظ محتويات النص و الآخر لتحميل محتويات النص ، (انظر الصورة على اليسار)، ما أريد القيام به هنا، حفظ قيمة هذا الصندوق على جهاز العميل ، ثم قراءة هذا الملف عند النقر على زر Load ، لنبدأ عملية الكتابة .



سنقوم بإنشاء طريقة لكتابة الملف و هي بسيطة جدا جدا إذا كنت تعلم كيف يتم التعامل مع Stream Write من قبل (تأكد من استدعاء فضاء الأسماء system.IO و System.IO.IsolatedStorage) ، إليك الطريقة :

```
1:  private void SaveData(string data, string fileName)
2:  {
3:  using (IsolatedStorageFile isf =
4:  IsolatedStorageFileStream(fileName, FileMode.Create, isf))
6:        {
7:  using (StreamWriter sw = new StreamWriter(isfs))
8:       {
9:       sw.Write(data);
10:      sw.Close();
11:       }
12:    }
13: }
```

حسنا لا تنصدم من طول الشفرة فكل ما في الأمر أني أكتب ملف له اسم يتم تمريره و لديه بيانات يتم تمريرها أيضا، ثم أنشأ كائن من IsolatedStorageFile و الذي بدوره يقوم بإنشاء الكائن StreamWriter حيث يتم الكتابة ، يمكن العودة إلى موقع MSDN للمزيد من المعلومات عن StreamWriter،

الخطوة التالية هي تنفيذ هذه الطريقة ، قم بإنشاء حدث Click للأداة Button و استدعي الطريقة ومرر لها البيانات الموجودة في صندوق النص مع اسم الملف و ليكن Test.txt ، الشفرة :




```
1:  private void button1_Click(object sender,
2:  RoutedEventArgs e)
3:    {
4:        SaveData(textBox1.Text, "test.txt");
5:    }
```

شغل التطبيق و جرب الأمر ، لن ترى شيء يحدث ، و لكن إن ذهبت إلى المسار التالي :
C:\Users\<UserName>\AppData\LocalLow\Microsoft\Silverlight\is

ثم بعد مجلد is ابحث عن مجلد التطبيق الخاص بك، ستجد ملف test.txt الذي قمنا بكتابته و به النص الذي قمت بكتابته في TextBox (انظر الصورة على اليسار) .

لاحظ أن مكان كتابة الملف يختلف من جهاز إلى أخر و لا تستطيع الحصول على مسار موحد لذلك عليك أن تبحث قليلا عن الملف في مجموع المجلدات الموجودة داخل is .



عند بحثك عن المسار المذكور أعلاه قد تحتاج إلى عملية اظهار مجلد AppData ، حيث يكون هذا المجلد مخفي افتراضيا ، لذلك يجب عليك اظهاره أولا .



الآن لنحاول القراءة من هذا الملف ، قم بكتابة الطريقة التالية :

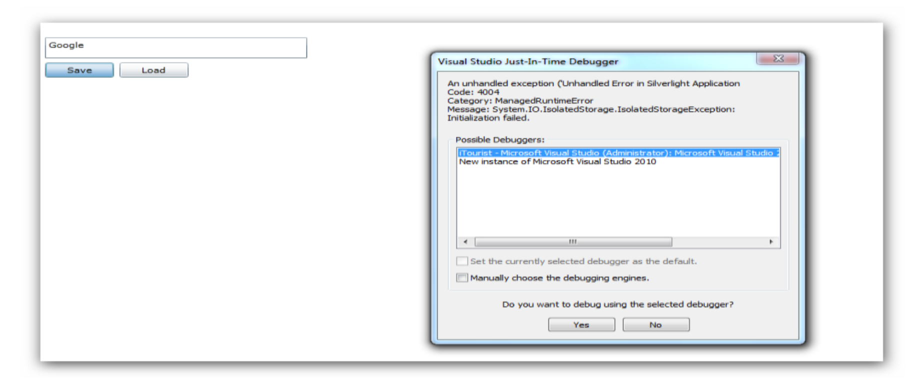
```
1:  private string LoadData(string fileName)
2:  {
3:      string data = String.Empty;
4:      using (IsolatedStorageFile isf =
5:      IsolatedStorageFile.GetUserStoreForApplication())
6:      {
7:      using (IsolatedStorageFileStream isfs = new
8:  IsolatedStorageFileStream(fileName, FileMode.Open, isf))
9:          {
10:       using (StreamReader sr = new StreamReader(isfs))
11:             {
12: string lineOfData = String.Empty;
13: while ((lineOfData = sr.ReadLine()) != null)
14: data += lineOfData;
15:             }
16:       }
17: }
18:             return data;
19: }
```

هذه المرة لدينا StreamReader بدلا من StreamWriter أيضا تعديل الطريقة String بدلا من كونها void ، انشأ حدث Click في أداة Button المسمى Load و استدعي الطريقة ، ثم مرر لها اسم الملف و اجعل القيمة المعادة توضع في صندوق النص ، الشفرة :

```
1:  textBox1.Text =  LoadData("test.txt");
```

شغل التطبيق الآن و انقر على زر Load هذه المرة و انظر للنتيجة .

هنا لدينا مشكلتين لنبدأ مع الأولى ، قم بتشغيل التطبيق وانقر بزر الفأرة الأيمن ستظهر لك نافذة اعدادات Silverlight حدد التبويب Application storage ، ستلاحظ ملف Cookies الذي قمت بكتابته في القائمة ، أسفل هذا التبويب قم بإلغاء الخيار Enable application Storage ، هذه المرة لن يتم تنفيذ التطبيق ، عوضا عن ذلك سترى نافذة Debugger المشئومة ،(انظر الصورة في الأسفل) .



لحل هذه المشكلة ، تذكر دائما أن تضع أي محاولة للكتابة أو القراءة داخل جملة try – catch ، سأعيد طريقة الاستدعاء في العنصر Load مثلا ، بشكل التالي :

```
1: try
2:   {
3:      textBox1.Text = LoadData("test.txt");
4:   }
5: catch (Exception Error)
6: {
7: MessageBox.Show(Error.Message);
8: }
```



المشكلة الأخرى إذا لم يكن الملف في الأصل موجود ! و لحل هذه المشكلة يمكنك استخدام الطريقة FileExists ، مرر لها اسم الملف ، حيث تعيد هذه الطريقة true إذا كان الملف موجود أو false إذا كان الملف غير موجود ، بهذه الطريقة :

```
1: IsolatedStorageFile store;
2: store = IsolatedStorageFile.GetUserStoreForApplication();
3:     if (store.FileExists("test.txt"))
4:         {
5:          textBox1.Text = LoadData("test.txt");
6:          }
```

أما عند حذف الملف لنفترض في حالة أن المستخدم قرر نسيان بياناته مثلا ، يمكننا استخدام DeleteFile ، الشفرة :

```
1: IsolatedStorageFile store;
2: store = IsolatedStorageFile.GetUserStoreForApplication();
3:    if (store.FileExists("test.txt"))
4:       {
5:       store.DeleteFile("test.txt");
6:       }
```

رأينا في هذا الفصل كيف تتم تخزين الأمور في Silverlight ، و ما هي الأخطاء التي من الممكن أن تقع أثناء عملية الكتابة و القراءة و كيف تمت معالجة هذه الأخطاء ، استخدمنا في هذا الأمر الكائن IsolatedStorageFile و الذي يتيح لك خدمات انسانية جليلة .



الفصل الرابع :

الوصول لخدمات الويب

لقد رأينا كيف أن تطبيقات Silverlight عبارة عن تطبيقات يتم تنفيذها على جهاز العميل و ليس على جهة المستخدم ! ماذا يعني هذا ، بكل بساطة هذا يعني أنك لن تستطيع تنفيذ شفرات الاتصال بقاعدة البيانات مثلا مباشرة من خلال Silverlight لذلك عليك أن تضيف طبقة أخرى و هنا يأتي دور web service ، ما سنتحدث عنه في هذا الفصل عبارة عن مقدمة للوصول للفصل القادم وهو الحديث عن قواعد البيانات .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- إنشاء خدمة web service .
- الوصول لخدمات الويب عن طريق Silverlight .
- الحدث Async .
- التعامل مع الحدث completed .



حسنا لنبدأ ، سوف أبدأ مشروع جديد من Silverlight و ليكن اسم المشروع مثلا webserviceExml في صفحة Silverlight فقط قم بإضافة Button و TextBox في منتصف الصفحة مثلا ، بما أننا سنتعامل مع WebSercice ، إذا سننتقل إلى الطرف الأخر من المشروع هذه المرة وهو web project ، في مثالنا هنا سيكون اسم هذا المشروع webserviceExml.Web ، اضغط بزر الفارة الأيمن و اختر اضافة ثم إضافة ملف جديد من نوع WebService (يمكنك البحث من الأعلى عن WebService) وليكن اسم الملف الجديد SendingEmailService.asmx ثم اضغط على OK .

```
// To allow this Web Service to be called from script, using ASP.NET AJAX, uncomment the following line.
// [System.Web.Script.Services.ScriptService]
public class WebService1 : System.Web.Services.WebService
{

    [WebMethod]
    public string HelloWorld()
    {
        return "Hello World";
    }
}
```

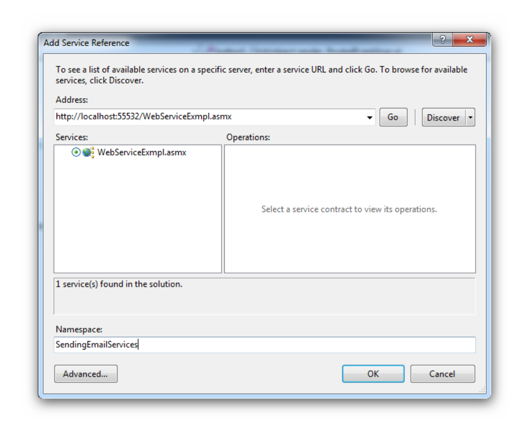
حسنا لنأتي إلى تفسير هذه الطلاسم ، لاحظ التعليق في الأعلى ، و الذي يخبرك بأنك إذا أردت الوصول إلى هذه الخدمة من خارج هذا التطبيق عليك تفعيل السطر الذي في الاسفل ، إذا الخطوة الأولى هي إزالة علامة التعليق لجعل هذه السطر من الكود ، واحذف الطريقة HelloWorld بما في ذلك كلمة [WebMethod]لأننا سنقوم بإنشاء طريقة تقوم بإرسال بريد الكتروني ، إليك شفرة الطريقة :

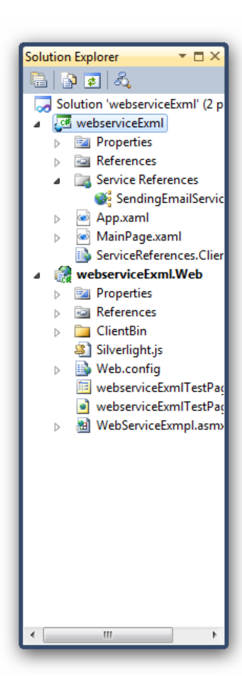



```
1:  public bool SendingEmail(string emailfor)
2:   {
3:   System.Net.Mail.MailMessage MyMailMessage = new
4:   System.Net.Mail.MailMessage("بريد المرسل", emailfor,
5:  "مرحبا من سيلفرلايت", "لوصولك كثيرا احيك العزيز صديقي
6:  هذه النقطة في الكتاب ، جميل جدا تابع إلى الامام");
7:  MyMailMessage.IsBodyHtml = false;
8:  System.Net.NetworkCredential mailAuthentication = new
9:  System.Net.NetworkCredential("بريدك الكتروني", كلمة المرور
10: "الخاصة بك");
:11 System.Net.Mail.SmtpClient mailClient = new
12: System.Net.Mail.SmtpClient("smtp.gmail.com", 587);
13: mailClient.EnableSsl = true;
14: mailClient.UseDefaultCredentials = false;
15: mailClient.Credentials = mailAuthentication;
16: Try
17: {
18: mailClient.Send(MyMailMessage);
19: return true;
20: }
21: catch { return false; }
22: }
```



تستقبل هذه الطريقة البريد الخاص بمستقبل الرسالة لاحظ أنك إذا أردت تنفيذ هذه الطريقة ستقوم بتغيير كلمة المرور و المرسل لهذه الرسالة ، أيضا لابد أن يكون البريد gmail لأنه مجاني فيما يخص stmp قم ببناء المشروع من القائمة Build أو من خلال CTRL + Shift + B ، بعد أن تنتهي من هذه التعديلات عد إلى Silverlight حدد مشروع Silverlight و انقر بزر الفأرة الأيمن و اختر الخيار add Service Reference ستظهر لك نافذة إضافة خدمات الويب (انظر الصورة على اليمين) ، انقر الزر Discover حتى يتم إيجاد الخدمة التي قمنا بعملها قبل قليل ، لاحظ أنك إذا لم تقم بتفعيل التعليق كما أخبرتك سابقا لن ترى هذه الخدمة في هذه الشاشة ، من صندوق النص بالأسفل غير اسم فضاء الأسماء إلى SendingEmailServices ، ثم اضغط على موافق ، سيقوم عندها VS بتوليد شفرة الخدمة بالكامل ، و لا يبقى لك سوى استخدام هذه الخدمة .





ستلاحظ إضافة مجلد جديد يحمل الاسم Service References و بداخله الخدمة الجديدة التي قمت ببنائها قبل لحظات (انظر الصورة على اليسار) ، الآن قم بالنقر على الـ Button الذي قمت بإضافته على الصفحة الرئيسية في البداية لتوليد الحدث click ، في أعلى الشاشة يجب عليك أولا أن تقوم باستدعاء مجال الأسماء الذي قمنا بإنشائه من خلال Add Service References ، في أعلى شاشة الشفرة أضف المشروع web كفضاء أسماء حتى تستطيع الوصول لهذه الخدمة التي قمت بإنشائها قبل قليل ، بالشكل التالي :

```
1: using webserviceExml.SendingEmailServices;
```

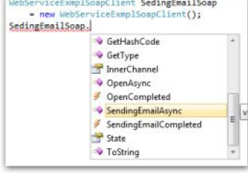
الآن توجه إلى الحدث Click ، ما سنقوم به في الشفرة التالية هو إنشاء كائن من النوع WebServiceExmplSoapClient من خلال هذا الكائن تستطيع الوصول لكل الطرق الموجودة في الخدمة التي قمت بإنشائها في مشروع الويب كما رأينا سابقا. قم بإنشاء كائن جديد من هذا النوع ، ثم حاول استخدامه للوصول للطريقة التي قمنا بكتابتها فيه ، ماذا تلاحظ ؟

صحيح لم تجد هذه الطريقة !! لماذا ؟ السبب هناك قانون ينص على الآتي " أي طريقة تريد الوصول لها موجودة في Web Service لابد وأن تكون مسبوقة بالوسم [WebMethod] ، لذلك لنعد لمشروع الويب و لنضف هذا الوسم فوق الطريقة التي قمنا ببنائها سابقا ، (انظر الصورة في الاسفل للتوضيح أكثر) :

```
[WebMethod]
public bool SendingEmail(string emailfor)
{
    System.Net.Mail.MailMessage MyMailMessage =
```

الآن أعد بناء التطبيق من جديد ، إذا حاولت مجددا الوصول لهذه الطريقة لن تفلح ، عليك أولا تحديث الخدمة في جهة Silverlight ، لذلك اذهب إلى مجلد Service References و حدد الخدمة التي قمنا بإنشائها سابقا انقر بزر الفأرة الأيمن و اختر Update Service References ، سيظهر أمامك معالج التحديث ، انتظر لحظات حتى الانتهاء .

بعد أن ينتهي معالج التحديث من تحديث الخدمة ، عد مجددا إلى الحدث Click و حاول الوصول للطريقة السابقة من خلال المتغير الذي قمنا بتعريفه سابقا ، هذه المرة سترى أنه بإمكانك الوصول للطريقة السابقة بسلام (انظر الصورة على اليسار)، و لكن الأمر مختلف قليلا ، دعني أشرح الفكرة ،





لكل طريقة تقوم ببنائها في ملف WebService يتم إنشاء طريقة للوصول لها تنتهي بكلمة Async ، وعندما تنتهي من تنفيذ هذه الطريقة نكتب بعدها حدث آخر ينتهي بالكلمة Completed . استدعى الطريقة SendingEmailAsync ومرر لها قيمة TextBox ، ثم قم بتفعيل الحدث Completed بعد أن تكتب SendingEmailCompleted خذ مسافة واحدة و اكتب علامة زائد و بعدها علامة يساوي ، ثم انقر على زر tab مرتين متتالية من لوحة المفاتيح مباشرة لكي يتم توليد شفرة الحدث بشكل آلي . من المفترض أن تكون شفرتك الآن بهذا الشكل :

```
private void button1_Click(object sender, RoutedEventArgs e)
{
    WebServiceExmplSoapClient SedingEmailSoap
        = new WebServiceExmplSoapClient();
    SedingEmailSoap.SendingEmailAsync(textBox1.Text);
    SedingEmailSoap.SendingEmailCompleted += new EventHandler<SendingEmailCompletedEventArgs>(SedingEmailSoap_SendingEmailCompleted);
}

void SedingEmailSoap_SendingEmailCompleted(object sender, SendingEmailCompletedEventArgs e)
{
    throw new NotImplementedException();
}
```

يقع الحدث Completed عندما تنتهي Silverlight من تنفيذ الطريقة SendingEmailAsync مباشرة ، أما عن النتائج المعادة فيتم وضعها في الخاصية result الخاصة بالمتغير e الخاص بالحدث Completed .

الآن لنكمل ما أريد أن أفعله هنا وهو إلغاء تفعيل TextBox و Button مباشرة بعد استدعاء الطريقة SendingEmailAsync ، ثم أعيد تفعيلهما بعد تنفيذ الطريقة ، مع إخبار المستخدم هل تم إرسال الرسالة بنجاح أم لا ، سأضع هذين السطرين بعد استدعاء الطريقة مباشرة لإلغاء تفعيل الأدوات :

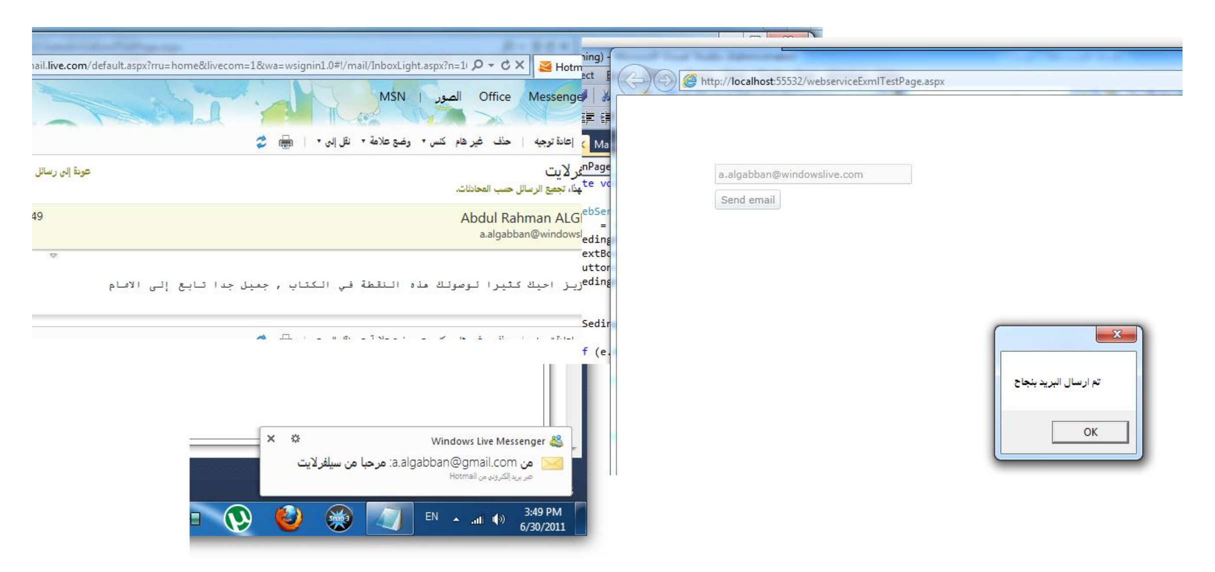



```
1: textBox1.IsEnabled = false;
2:  button1.IsEnabled = false;
```

و ذلك داخل الحدث Click ، نأتي للطرف الآخر من المعادلة ، إلى الحدث Completed الشفرة :

```
1: if (e.Result)
2:  {
3: MessageBox.Show(" تم ارسال البريد بنجاح ");
4:  }
5: else
6:  {
7: MessageBox.Show(" عفوا لم يتم ارسال البريد بنجاح :( ");
8:  }
9: textBox1.IsEnabled = true ;
10: button1.IsEnabled = true ;
```

لا أستطيع أن أشرح شيء في هذه الشفرة ، اضغط على F5 لتجربة التطبيق ، ادخل بريد المرسل إليه في TextBox ، من المفترض أن يعمل التطبيق بسلام ☺ .



حسنا لقد اجتزت خطوة جدا مهمة في تطبيقات Silverlight لا أبالغ إن أخبرتك بأن ما يقارب ٩٠٪ من التطبيق سيكون في مشروع Web و ليس في تطبيق Silverlight عندما يأتي الحديث لقواعد البيانات أو مثلا رفع الملفات أو إرسال لبريد إلكتروني ... الخ .



لقد تعرفت في هذا الفصل كيف تستطيع التواصل مع خدمات الويب ، و كيف تستطيع بناء واحدة و استخدامها ، و كيف تستطيع استيراد هذه الخدمة إلى تطبيقات Silverlight ، لقد تعمدت أن لا أكتب الوسم [WebMethod] حتى أجعلك تستخدم الخاصية Update Service References و التي فعلا تقدم لك خدمة جليلة جدا لن تستطيع أن تفيها حقها من الشكر . قم بتفحص البريد الذي أرسلت له الرسالة و إلى الخطوة التالية .



اذا أردت تجربة الشفرة المرفقة لهذا الفصل ، لا تنسى أن تقوم بتغيير البريد الالكتروني و كلمة المرور في الطريقة SendingEmail الموجودة في خدمة الويب .

الفصل الخامس :

الوصول لقواعد البيانات

لقد تعرفنا في الفصل السابق على خدمات الويب ، و رأينا كيف يتم التعامل مع خدمات الويب من خلال Silverlight ، في هذا الفصل سنستخدم ما تقدم في الفصل السابق لكي نرى كيف يتم الوصول لقواعد البيانات، من المفترض بعد تعرفك على خدمات الويب أن يكون الأمر هنا أكثر بساطة بكثير، فكل شيء عبارة عن WebService ، أيضا سنرى كيف يتم الاتصال بقاعدة البيانات من خلال WCF RIA بدلا من أن تقوم بعمل كل شيء بشكل يدوي و سترى كيف أن WCF RIA ستختصر الكثير من الوقت .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- استخدام Web Service .
- استخدام WCF RIA .



استخدام Web Service

حسنا الاتصال بقاعدة البيانات ليس بالأمر الصعب نهائيا في Silverlight ، هناك طريقتين للاتصال بقاعدة البيانات في Silverlight إما أن تترك كل شيء لـ WCF RIA أو أن تهتم أنت بكل شيء بنفسك ، سأتطرق هنا لكلتا الطريقتين في Silverlight ، لاحظ أيضا لن أتحدث عن ADO.net أو الاتصال بقواعد البيانات .. الخ ، بل سأنتقل مباشرة إلى التنفيذ لأني أفترض بأنك تعرف هذه المعلومات سابقا ، لدينا جدول في قاعدة البيانات بالشكل التالي :

Column Name	Data Type	Allow Nulls
ID	int	☐
Name	nvarchar(50)	☐
Email	nvarchar(50)	☐
Password	nvarchar(50)	☐
webSite	nvarchar(50)	☐
		☐

و لدينا به مجموعة من البيانات أياً تكن ، ما نريده الآن هو الحصول على تلك البيانات من Silverlight ، طبعا سأفترض أنك قمت بإنشاء تصنيف باسم Users في مشروع web للحصول على الاتصال و من ثم طريقة للحصول على جميع القيم في قاعدة البيانات من هذا الجدول ، إليك الشفرات :




```
1:  public class UserDAL
2:    {
3:  static string connectionString =
    Connections.connectionString;
4:  public List<Users> GetUsers()
5:  {
6:  SqlConnection Connection = new
    SqlConnection(connectionString);
7:  DataSet objSet = new DataSet();
8:  List<Users> lstResult = null;
9:  SqlCommand Command = new SqlCommand();
10: Command.Connection = Connection;
11: Command.CommandText = "LoadingUsers";
12: Command.CommandType =
    CommandType.StoredProcedure;
13: SqlDataAdapter sqlDataAdapter = new
    SqlDataAdapter();
14: sqlDataAdapter.SelectCommand = Command;
15: Try
16: {
17: sqlDataAdapter.Fill(objSet);
18: Users objUsers;
19: if (objSet.Tables.Count > 0)
20: {
21: lstResult = new List<Users>();
22: foreach (DataRow dr in objSet.Tables[0].Rows)
23: {
24:  objUsers = new Users();
25: objUsers.id = int.Parse(dr["ID"].ToString());
```




```
26: objUsers.name = dr["Name"].ToString();
27: objUsers.email = dr["Email"].ToString();
28: objUsers.passwrod = dr["Password"].ToString();
29: objUsers.webSite = dr["webSite"].ToString();
30: lstResult.Add(objUsers);
31: }
32: }
33: }
34: catch { }
35:   finally
36: {
37: Command.Connection.Close();
38: Command.Parameters.Clear();
39: Command.Dispose();
40: sqlDataAdapter.Dispose();
41: }
42: return lstResult;
43: }
44: }
45:
46: public class Connections
47: {
48: public static string connectionString =
    ConfigurationManager.
49: ConnectionStrings["ConnectionStr"].ConnectionString;
50: }
51:
52: public class Users
53: {
```




```
54:  public int id { get; set; }
55:  public string name { get; set; }
56:  public string email { get; set; }
57:  public string passwrod { get; set; }
58:  public string webSite { get; set; }
59:  }
```

الآن لنستخدم ما تحدثنا عنه في الفصل السابق و هو Web Service لتمرير هذه الطريقة لـ Silverlight ، لعمل ذلك سنقوم بإضافة web Service جديدة و سمها كما شئت مثلا databaseSr،فقط قم بإنشاء طريقة جديدة و ليكن اسمها LoadingUsers على أن تعيد الاخيرة Collections List من النوع Users ، إليك الشفرة :

```
1:  [WebMethod]
2:  public List<Users> LoadingUsers()
3:    {
4:     UserDAL usr = new UserDAL();
5:     return usr.GetUsers();
6:    }
```

لنعد إلى Silverlight اسحب data grid إلى النموذج ، ثم أضف Service References جديدة تشير إلى تلك التي قمت بعملها قبل قليل كما رأينا في الفصل السابق لا تنسى طبعا أن تقوم ببناء التطبيق قبل استدعاء الخدمة ،



في المشيد الرئيسي للصفحة لن نقوم بعمل أي شيء جديد هنا ☺ فقط سنقوم باستدعاء الطريقة السابقة و جعل البيانات كـ ItemsSource للأداة Data grid تمام كما رأينا في فصل data binding ، اليك الشفرة :

```
1:  public MainPage()
2:  {
3:   InitializeComponent();
4:   ServiceReference.databaseSrSoapClient SD
5:              = new databaseSrSoapClient();
6:   SD.LoadingUsersAsync();
7:   SD.LoadingUsersCompleted += new EventHandler
8:   <LoadingUsersCompletedEventArgs>(SD_LoadingUsersCompleted);
9:      }
10:  void SD_LoadingUsersCompleted(object sender,
11:      LoadingUsersCompletedEventArgs e)
12:    {
13:    if (e.Result != null)
14:     {
15:    dataGrid1.AutoGenerateColumns = true;
16:    dataGrid1.ItemsSource = e.Result;
17:     }
18: }
```

شغل التطبيق الآن حتى ترى البيانات .

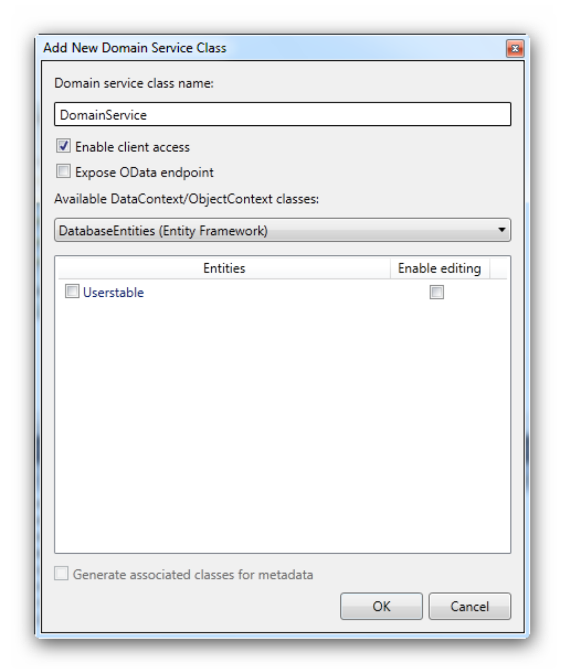


استخدام WCF RIA .

لنرى الآن الطريقة الأخرى للاتصال بقاعدة البيانات في تطبيقات Silverlight و ذلك باستخدام WCF RIA افتح مشروع جديد و لكن هذه المرة توقف قليلا عند شاشة البداية ، قم بتفعيل الخيار Enable WCF RIA Services ثم انقر على موافق .

الخطوة التالية حدد مشروع asp.net ثم قم بإضافة ADO.net Entity ، طبعا هنا أفترض أن لديك قاعدة البيانات) يمكنك أن تعثر على قاعدة بيانات هذا المثال في مجلد Database < Chapter 5 (أعد التسمية إلى Users مثلا في مثالنا هنا ، سيطلب منك المعالج في الخطوة الأولى اذا كانت البيانات من DB أو من XML حدد DB إذا كنت قد قمت بإضافة قاعدة البيانات للمشروع سيعثر عليها المعالج ، أو حدد أي قاعدة بيانات ترغب لا يهم ذلك كثيرا ، المهم أن تتصل بقاعدة بيانات أي كانت . ثم اضغط التالي ، حدد الجداول اضغط انتهاء .





الخطوة الثانية هيا إضافة Domain Service لكن أولا قم ببناء المشروع ثم من جديد ابقى في مشروع ASP.net و أضف عنصر جديد من النوع Domain Service Class ، في الصورة على اليسار سترى خيارين ما يهمنا هو الخيار الأول حيث يتيح لك هذا الخيار إمكانية الوصول لتطبيقات Client لهذا Domain تأكد من تحديدك لهذا الخيار ثم اختر الجدول الذي تريد عرض بياناته على Silverlight في حالتنا هنا سيكون userstable أيضا لاحظ يمكنك تمكين عمليات التعديل على الجدول من خلال تحديد الخيار enable editing حيث سيقوم هذا الخيار بتوليد شفرات التحديث و الإضافة و الحذف ، تأكد من تحديدك أيضا الخيار الذي بالأسفل المسمى Generate associated classes from metadata ، اضغط على موافق الآن .



اعد بناء التطبيق الآن وعد إلى تطبيق Silverlight ، أضف في الصفحة الرئيسية أداة data grid ، افتح شفرة الصفحة الرئيسية ، ثم أضف الشفرة التالية :

```
1: DomainService1 context = new DomainService1();
2: context.Load<Userstable>(context.GetUserstableQuery());
```

السطر الأول قمت بإنشاء كائن من الخدمة التي قمت بإنشائها في تطبيق asp.net لاحظ هنا أني قمت باستدعاء المشروع web كفضاء أسماء في الأعلى حتى أستطيع الوصول لهذه الخدمة التي قمت بإنشائها قبل قليل ، الشفرة :

```
1: using databaseUsingWCFRIA.Web;
```

في السطر الثاني قمت باستدعاء الطريقة GetUserstableQuery ووضع جميع النتائج داخل الكائن Context على أن تكون هذه النتائج من النوع Usertable ، ما تبقى هو العودة بهذه القيم إلى الأداة data grid ، الشفرة :

```
1: dataGrid1.AutoGenerateColumns = true;
2: dataGrid1.ItemsSource = context.Userstables;
```



شغل التطبيق الآن و انظر للنتيجة ، رأينا في هذا الفصل كيف يتم التواصل بين تطبيقات Silverlight و قواعد البيانات ، الأمر لا يختلف إطلاقا عند الحديث عن عمليات التحديث أو الإضافة أو الحذف يمكنك استخدام إحدى الطريقتين إما يدويا أو استخدام ADO.net مع WCF RIA ، الخيار لك في النهاية .



يمكن الحصول على معلومات أكثر عن هذه التقنيات من خلال هذه الروابط :

- WCF RIA Services
- riaservices



الفصل السادس :

التعامل مع الوسائط المتعددة

سنتحدث في هذا الفصل عن كيفية التعامل مع الوسائط المتعددة مثل ملفات الأصوات و الفيديو من خلال Silverlight كما سنتطرق إلى التعامل مع كاميرا الويب .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- أداة Media Element .
- استخدام كاميرا الويب .



استخدام الوسائط المتعددة أمر مهم جدا في عالم الويب ، لعرض مقاطع الفيديو مثلا و استخدام كاميرا الويب أو عرض الصور و تشغيل ملفات الصوت ، أحد أهم الأدوات التي سنستخدمها للتعامل مع الوسائط المتعددة الأداة MediaElement .

استخدام اداة MediaElement :

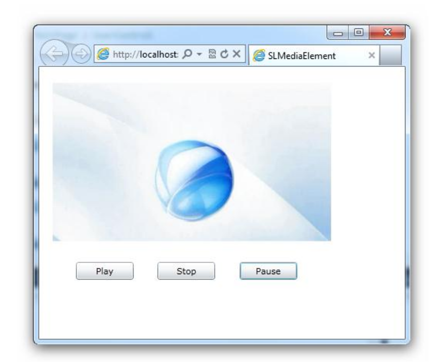
أحد أهم الأمور التي يجب أن تعرفها هي أن أداة MediaElement تتعامل فقط مع هذا النوع من ملفات الفيديو WMV من الإصدار الأول حتى الإصدار الثالث منه و أيضا WMVA و WMVAC و أخيرا H.264 ، أما عن ملفات الصوت WMA من الإصدار السابع حتى التاسع و أيضا ملفات MP3 ، وتدعم هذه الأداة أيضا البث المباشر .

لنبدأ العمل قم بإنشاء مشروع جديد من Silverlight في الواجهة الرئيسية حدد الأداة MediaElement ستجد هذه الأداة في القسم All Silverlight Controls من شريط الأدوات ، لن ترى شيء جديد حتى اللحظة ستجد أداة فارغة على الصفحة ، لتحديد مقطع فيديو أو صوت ستستخدم الخاصية Source بإمكانك عمل هذا الأمر من خلال شفرة XAML حتى نتطرق لما يسمى بمصادر المشروع ،



أولا أضف ملف الفيديو المرفق مع المشروع أو أي ملف فيديو أخر إلى المشروع (إلى تطبيق Silverlight) حدد هذا الملف ثم انقر بزر الفأرة الأيمن و حدد الخيار properties تأكد من تغيير الخاصية Build Action إلى Rescues ، ثم من شفرة XAML أضف الخاصية source مع تعديل قيمتها إلى مسار ملف الفيديو بهذا الشكل :

```
1:  <MediaElement Source="Silverlight.wmv"
2:    Height="213" HorizontalAlignment="Left"
3:    Margin="17,21,0,0" Name="mediaElement1"
4:    VerticalAlignment="Top"  Width="364" />
```



الآن شغل التطبيق و سيبدأ الفيديو بالعمل مباشرة ، خطوتنا التالية هيا التحكم بهذا الفيديو من التوقف المؤقت و التشغيل و التوقف كليا ، أضف ثلاثة Button للواجهة ثم انشأ حدث Click لكل عنصر منهم و اكتب الشفرات التالية في الزر المخصص له :

```
1:  private void button1_Click(object sender, RoutedEventArgs e)
2:    {
3:   mediaElement1.Play();
4:    }
5:
```




```
6:   private void button2_Click(object sender, RoutedEventArgs e)
7:    {
8:  mediaElement1.Stop();
9:    }
10:
11:  private void button3_Click(object sender, RoutedEventArgs e)
12:   {
13: mediaElement1.Pause();
14:   }
```

أعد تشغيل التطبيق و جرب الأزرار ، بهذا نكون قد أنجزنا تقريبا ٣٠٪ من العمل لكن ما زال أمامنا المزيد ، في الخطوة التالية سنضيف شريط Slider ليعرض مدى تقدم الفيديو .

لعمل ذلك سنحتاج للحدث MediaOpended للحصول على طول الفيديو أولا ، في البداية حدد الأداة Slider من شريط الأدوات ثم أضفها للنموذج ، انشأ الحدث MediaOpended و ذلك بتحديد العنصر MediaElement و من ثم التوجه لنافذة الخصائص و في تبويب الأحداث انقر نقرا مزدوجا على الحدث MediaOpended ، للحصول على طول الفيديو سنحتاج للخاصية NaturalDuration التابعة للأداة MediaElement و التي بدورها لديها خاصية اسمها TimeSpan ، والآن اكتب شفرة الحدث MediaOpended :




```
1: slider1.Maximum =
2: mediaElement1.NaturalDuration.TimeSpan.TotalSeconds;
```

بعد أن حصلنا على طول المقطع و جعلنا القيمة العظمى للأداة Slider تساوي هذه القيمة ، نحتاج أيضا أن نغير هذه الخاصية value للأداة Slider أثناء تشغيل الفيديو ، لعمل ذلك سنضطر إلى إنشاء timer لكي يؤقت هذه العملية ، في أعلى الشفرة قبل سطر MainPage () أضف السطر التالي :

```
1: System.Windows.Threading.DispatcherTimer videoTimer
2:    = new System.Windows.Threading.DispatcherTimer();
```

في المشيد الرئيسي للصفحة سنفعل الخاصية Interval إلى ثانية ، الشفرة :

```
1: videoTimer.Interval = new TimeSpan(0, 0, 0, 1);
```

ثم سنطلق الحدث Tick ، اكتب =+ videoTimer.Tick ثم انقر على Tab مرتين ، سينشأ VS الحدث videoTimer_Tick داخل هذا الحدث سنقوم بتعديل الخاصية value للعنصر Slider إلى الثانية التي يعرض بها المقطع حاليا ، و لكن قبل ذلك سنجعل videoTimer يبدأ بالعمل باستخدام الطريقة start بشكل التالي :




```
1:  videoTimer.Start();
```

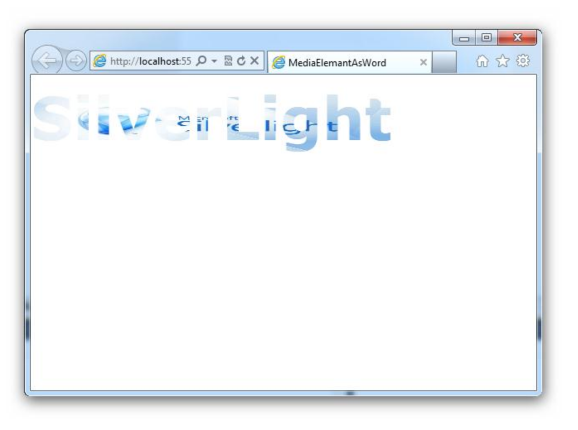
ليصبح المشيد بهذا الشكل :

```
1:  public MainPage()
2:    {
3:   InitializeComponent();
4:   videoTimer.Interval = new TimeSpan(0, 0, 0, 1);
5:   videoTimer.Tick += new EventHandler(videoTimer_Tick);
6:   videoTimer.Start();
7:    }
```

الآن في الحدث videoTimer_Tick اكتب الآتي :

```
1:  slider1.Value = mediaElement1.Position.Seconds;
```

الآن قم بتشغيل التطبيق لترى شريط Slider يعمل .



هناك المزيد أيضا ، يمكنك استخدام الفيديو الذي يتم عرضه على هذه الأداة لتشكيل كلمات و حروف أو ما شابه ذلك !! ، كما في الصورة على اليسار ، لنبدأ مشروع جديد في Silverlight ، ثم أضف ملف الفيديو و اجعله كـ Rescues ، أضف في الصفحة الرئيسية أداة MediaElement و أداة TextBlock ،



قم بتصغير حجم أداة MediaElement حتى لا يمكن رؤيتها و وتكبير الأداة TextBlock على كامل الصفحة ، قم بتغيير الخاصية FontSitze إلى ٧٢ ، الخدعة في الأمر هو تغيير خلفية الكلمة و ليس تشكيل كلمة من مقطع فيديو ☺ .

لعمل ذلك نحتاج إلى ما يسمى بـ videoBrush و الخاصية Foreground الخاصة بـ TextBloxk ، حيث سوف يتم تغيير الخاصية SourceName التابعة لـ VideoBrush إلى اسم الأداة MediaElement بالشكل التالي:

```
1: <TextBlock Height="203" HorizontalAlignment="Left"
2: Margin="10,26,0,0" Name="textBlock1" Text="TextBlock"
3:  VerticalAlignment="Top" FontSize="72" Width="378">
4:   <TextBlock.Foreground>
5:  <VideoBrush SourceName="mediaElement1">
6:  </VideoBrush>
7:   </TextBlock.Foreground>
:8  </TextBlock>
```

قم بتشغيل التطبيق الآن .



استخدام كاميرا الويب .

في البداية لنقم بإنشاء مشروع جديد في Silverlight أضف إلى النموذج أداة Button وأداة StackPanel داخل StackPanel أضف أداة Rectangle ، كم تتخيل حجم الشفرة التي ستستخدمها للتعامل مع كاميرا الويب !! مقارنة بـ Adobe Flash لا شيء يذكر ☺ ، انشأ حدث Click لـ Button ، في البداية علينا التأكد من وجود كاميرا لدى المستخدم قبل محاولة استخدامها ، و أخذ الأذن من المستخدم للتعامل مع كاميرا الويب ، يكفي فقد تعريف متغير من نوع Bool للحصول على هذا الأذن ، في الحدث Click اكتب التالي :

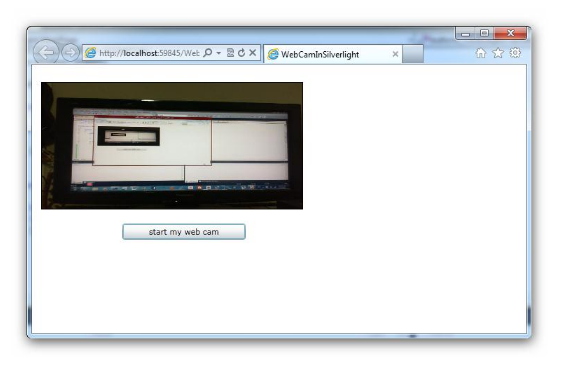
```
1: bool b =
2: CaptureDeviceConfiguration.RequestDeviceAccess();
```

قم بتشغيل التطبيق الآن و انقر على Button ستجد بأن Silverlight يطلب الأذن من المستخدم للوصول للأجهزة المتصلة بجهاز العميل ، إليك الشفرة للحصول على الكاميرا ثم سنأتي للشرح :

```
1: bool b = CaptureDeviceConfiguration.RequestDeviceAccess();
2:   if (b)
3:  {
4:  CaptureSource webCam = new CaptureSource
5: {
```




```
6:  VideoCaptureDevice =
7: CaptureDeviceConfiguration.GetDefaultVideoCaptureDevice(
8: )};
9: VideoBrush vidB = new VideoBrush();
10: vidB.SetSource(webCam);
11: rectangle1.Fill = vidB;
12: webCam.Start();
13: }
```



بعد التأكد من أن المستخدم قام بمنحي الصلاحيات للوصول إلى كاميرا الويب ، أقوم بإنشاء عنصر من النوع CaptureSource و الذي بدوره يحدد كاميرا الويب الافتراضية على جهاز العميل ثم أقوم بإنشاء كائن من النوع VideoBrush و الذي يستخدم لاحتواء مقاطع فيديو كما رأينا سابقا حيث سيكون مصدر الفيديو الكائن WebCam ، أخيرا أقوم بتعبئة هذا الفيديو داخل أداة Rectangle ثم ابدا عرض الويب .

هناك العديد من الطرق التي يمكنك الاستفادة داخل الكائن WebCam مثلا يمكنك تسجيل الأصوات أثناء عمل الكاميرا من خلال وضع الخاصية AudioCaptureDevice إلى



CaptureDevicesConfiuration.GetDefultAudioCaptureDevices ،

أيضا يمكنك التوقف من خلال الطريقة Stop .

قبل أن نختم هذا الجزء من Silverlight لنقم بعمل تطبيق باستخدام Web Cam في البداية تخيل لدينا السيناريو التالي ، أثناء تسجيل الدخول ، تطلب من المستخدم صورة معينة حيث تتيح له استخدام كاميرا الويب لالتقاط صورة له ، يقوم التطبيق بالتقاط الصورة على الكاميرا و تحويلها إلى صورة من النوع jpg لنبدأ ، في البداية قم بتصميم نموذج بسيط يحتوي على زرين الأول start web cam و الآخر take this as an image طبعا لدينا أداة Rectangle داخل أداة Grid هذه المرة ، الآن من شفرة XMAL يجب أن تتأكد من rectangle تملئ تماما أداة grad الجديدة (لحاجة في نفس يعقوب) ، أما عن 'Start Button web cam' قمنا بعمل شفرته سابقا ، سننتقل للخطوة التالية ،

و هي شفرة Button 'take this as an image' ، في البداية ستجد في المرفقات مكتبة بعنوان FJ.Core أضفها كمرجع لمشروع Silverlight ثم قم باستدعاء فضائي الأسماء الموجودة في هذه المكتبة :

```
1: using FluxJpeg.Core;
2: using FluxJpeg.Core.Encoder;
```



لأننا سنحفظ الصورة على مكان يحدده المستخدم سنحتاج أيضا إلى فضاء الأسماء IO و أيضا فضاء الأسماء Windows.Midea.Imaging ، بهذا الشكل :

```
1: using System.IO;
2: using System.Windows.Media.Imaging;
```

في أعلى الشفرة سنقوم بتعريف كائن من النوع SaveFileDialog ، ثم في المشيد الرئيسي سنقوم بتعيين هذا المتغير و تحديد أنواع الملفات ، الأمر تماما كما هو عليه في Windows Form ، الشفرة :

```
1: private SaveFileDialog saveFileDlg;
2: public MainPage()
3:  {
4:  InitializeComponent();
5: saveFileDlg = new SaveFileDialog
6:       {
7:  DefaultExt = ".jpg",
8: Filter = "JPEG Images (*jpeg *.jpg)|*.jpeg;*.jpg",
9:       };
10: }
```

الخطوة التالية من العمل هي استخدام المكتبة FJ.Core لتحويل الصورة التي تم التقاطها إلى FluxJpeg ثم أخيرا تحويلها إلى jpegImage يمكنك مراجعة مكتبة MSDN للمزيد من التفاصيل :

```
1: public static void EncodeJpeg(WriteableBitmap bmp,
2: Stream dstStream)
```




```
3:  {
4:   int w = bmp.PixelWidth;
5:   int h = bmp.PixelHeight;
5:   int[] p = bmp.Pixels;
6:   byte[][,] pixelsForJpeg = new byte[3][,];
7:   pixelsForJpeg[0] = new byte[w, h];
8:   pixelsForJpeg[1] = new byte[w, h];
9:   pixelsForJpeg[2] = new byte[w, h];
10:  int i = 0;
11:  for (int y = 0; y < h; y++)
12:      {
13:   for (int x = 0; x < w; x++)
14:       {
15:   int color = p[i++];
16:   pixelsForJpeg[0][x, y] = (byte)(color >> 16); // R
17:   pixelsForJpeg[1][x, y] = (byte)(color >> 8);  // G
18:   pixelsForJpeg[2][x, y] = (byte)(color);       // B
19:       }
20:      }
21: var jpegImage = new FluxJpeg.Core.Image(new ColorModel
22: { colorspace = ColorSpace.RGB }, pixelsForJpeg);
23: var encoder = new JpegEncoder(jpegImage, 95, dstStream);
24: encoder.Encode();
25: }
```

تستقبل الطريقة كائن من النوع writeableBitmp و كائن من النوع Stream

الأول يمثل الصورة والثاني يمثل مكان حفظ الصورة ، لا توجد طريقة مباشرة لأخذ



الصورة من الكاميرا ، هناك عملية تحايل ما سنقوم به هو أخذ صورة لـ Grid التي تحتوي على Rectangle و بما أن الأخيرة ممتدة تماما على grid فهذا يعني أن المستخدم سيحصل على الصورة التي التقطها من الكاميرا ، في البداية علينا تحويل grid إلى كائن من النوع writeableBitmp و من ثم نحدد المكان الذي يريد المستخدم حفظ الصورة فيه ، ثم نقوم باستدعاء الطريقة التي في الأعلى ، طريقة saveWebCam :

```
1:  private void SaveWebCam(Stream dstStream)
2:  {
3:    try
4:    {
5:   WriteableBitmap bmp = new WriteableBitmap(grid1, null);
6:   EncodeJpeg(bmp, dstStream);
7:    }
8:   catch (Exception ex){
9:    MessageBox.Show(ex.Message, "Error saving webCam",
10: MessageBoxButton.OK);
11: }
12: }
```

لم يتبق الآن سوى استدعاء هذه الطريقة مع تمرير مكان الحفظ ، في الزر Take this as image استدعي الطريقة السابقة بهذا الشكل :




```
1: if (saveFileDlg.ShowDialog().Value)
2:    {
3:    using (Stream dstStream = saveFileDlg.OpenFile())
4:            {
5:          SaveWebCam(dstStream);
6:            }
7:    }
```

شغل التطبيق ثم اضغط على زر تشغيل الكاميرا ، و أخيرا زر التقاط الصورة.

إلى هنا نكون قد انتهينا من هذا الجزء ، رأينا فيه كيف يتم التعامل مع خدمات الويب في Silverlight، و كيف يتم الاتصال بقاعدة البيانات يدويا وباستخدام المعالج ، ثم رأينا كيفية عرض مقاطع الفيديو باستخدام الأداة MediaElement ،ورأينا أيضا كيف يتم التعامل مع كاميرا الويب وأخيرا استخدام IsolatedStorageFile لتخزين ملفات صغيرة مثل ملفات Cookies على جهاز العميل .

٣

الجزء الثالث

المزيد عن Silverlight

في هذا الجزء ستتعرف على العديد من الأمور الأكثر تطويرا في Silverlight ، التعرف على أجزاء المشروع و تشغيل التطبيقات خارج المتصفح ، ثم التعامل مع الثيمات ، ثم سنتطرق أخيرا لنقطة مهمة وهي العمل خارج المتصفح

بعد الانتهاء من هذا الجزء يجب أن يضاف إلى معرفتك التالي :

- أجزاء مشاريع و مجلدات Silverlight .
- التعامل مع تطبيقات Silverlight خارج المتصفح .



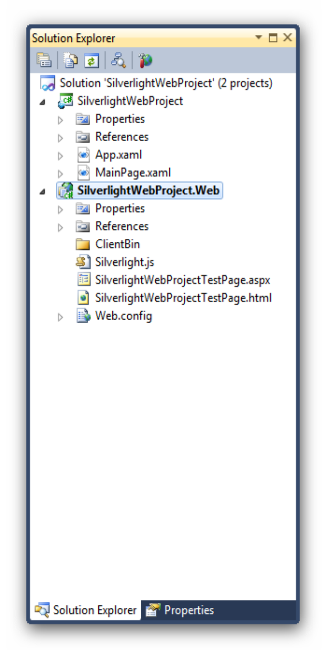
الفصل السابع :

أجزاء مشاريع Silverlight

كان يجب أن أتطرق لهذه النقطة في بداية الكتاب ، و لكن رغبت بأن أبدأ بالعمل مباشرة حتى أبعد عنك الملل قليلا ، في هذا الفصل سترى ما هي المجلدات الرئيسية و كيف يتم الانتقال بين الصفحات في تطبيقات Silverlight .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- مكونات مشروع الـ web .
- مكونات المشروع Silverlight .
- الملف app.xaml .
- الانتقال بين الصفحات .



مكونات مشروع الـ Web

وصولك لهذه النقطة يحتم عليك أن تعرف كيف تتم الأمور في الخلف ، لقد تحدثنا في المقدمة بأن تطبيقات Silverlight يتم تخزينها على جهاز العميل و لكن أين يتم تخزينها على جهاز الخادم ! لكي يقوم العميل بتحميلها ، سوف تساعدك هذه النقطة كثيرا عند الحديث عن نشر تطبيقات Silverlight إلى الويب .

إذا ألقيت نظرة على مشروع الـ Web ستجد مجلد اسمه ClientBin و في هذا المجلد يتم تخزين مشروع Silverlight الذي ينتهي بالامتداد xap. على جهة الخادم ، السؤال الآن كيف يعرف الخادم مسار التطبيق ! هنا يأتي دور صفحة Aps.net أو HTML فكلا الصفحتين تحتوي على Form هذا الـ Form لديه وسم اسمه source و هذا الأخير يخبر صفحة الويب بأن تطبيق Silverlight موجود في هذا المسار .

```
<form id="form1" runat="server" style="height:100%">
<div id="silverlightControlHost">
    <object data="data:application/x-silverlight-2," type="application/x-silverlight-2" width="100%" height="100%">
      <param name="source" value="ClientBin/SilverlightWebProject.xap"/>
      <param name="onError" value="onSilverlightError" />
      <param name="background" value="white" />
      <param name="minRuntimeVersion" value="4.0.50826.0" />
      <param name="autoUpgrade" value="true" />
      <a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.50826.0" style="text-decoration:none">
          <img src="http://go.microsoft.com/fwlink/?LinkId=161376" alt="Get Microsoft Silverlight" style="border-style:none"/>
      </a>
    </object><iframe id="_sl_historyFrame" style="visibility:hidden;height:0px;width:0px;border:0px"></iframe></div>
</form>
```



حسنا ماذا إذا لم يستطع هذا Form إيجاد التطبيق في مجلد ClientBin عندها يطلق حدث يسمى OnError ، لماذا هناك صفحتين واحدة HTML و الأخرى ASP.net ، بكل بساطة إذا كنت ستجري شفرات من جهة الخادم فعليك استخدام الصفحة asp.net و بعض أدوات Silverlight لابد أن يتم استدعائها من خلال asp.net و ليس HTML كما هو الحال مع أداة Bing Map .

بقى لدينا ملف واحد من جهة الخادم وهو Silverlight.js هذا الملف يمكن اعتباره أداة لاحتواء تطبيق Silverlight و عرضها على الصفحة الرئيسية .

مكونات مشروع Silverlight

من جهة أخرى تطبيق Silverlight يمكن أن يحتوي على العديد من الملفات بداخله و لكن سأتحدث عن المجلدات الرئيسية فقط ، المجلد الأول هو Service References ويحتوي هذا المجلد على جميع خدمات الويب التي قمت باستدعائها من مشروع الويب أو من أي جهة أخرى .



لديك ملف لا يخلو أي تطبيق في Silverlight منه وهو app.xaml و هو يوازي تماما program.cs في تطبيقات windows form سنتحدث عن هذا الملف في النقطة التالية ، أخيرا لديك مجلدين هما references و Properties الأول يحتوي على المراجع التي أضفتها للمشروع و الثاني يحتوي على ملف setting الخاص بالمشروع لاحظ هنا شيء مهم جدا ، بما أن تطبيقات Silverlight تعمل على جهة المستخدم فإن ملف Setting التابع لتطبيق Silverlight يتم تخزينه على جهاز العميل و هو يختلف تماما عن ملف setting التابع لتطبيق الـ web .

الملف App.xaml

كما رأيت في الصفحة السابقة ، يعتبر هذا هو نقطة البداية لتطبيقات Silverlight على عكس ملفات XAML الموجودة لديك في المشروع ، ليس لدى app.xaml واجهة رسومية ، كما أنه ينقسم إلى قسمين الشق الأول من هذا الملف ينتهي بامتداد xaml و الثاني ينتهي بامتداد cs ، يستخدم الشق الأول دائما لتعريف الملفات أو الأدوات و ما شابه ذلك التي تريدها أن تكون كمصدر للمشروع بحيث يمكن أن يصل لها أي جزء من التطبيق (سنرى هذا الأمر في فصل التعامل مع Themes) ، أما الشق الأخر و الذي ينتهي بامتداد cs فهو محور حديثنا في هذا الفصل .



لدى هذا الملف مشيد و أربعة طرق مبينة في الجدول التالي :

اسم الطريقة	الوظيفة
Application_Startup	و من هنا يتم تحديد نقطة الانطلاق في التطبيق أيضا يمكنك هنا إنشاء بعض الطرق لكي يتم تنفيذها مباشرة بعد أن يتم تحميل التطبيق على جهاز العميل .
Application_Exit	على عكس الأولى يتم إطلاق هذا الحدث مباشرة عند إغلاق التطبيق .
Application_UnhandleException	عند حدوث أية أخطاء لم تحاول اصطيادها يتم إطلاق هذا الحدث مباشرة ، لذلك يستخدم مطوري الـ Silverlight هذا الحدث عند وقوع خطأ بتحديث الصفحة مباشرة .
ReportErrorToDOM	في حال وقوع أخطاء تمت معالجتها أو اصطيادها إن صح التعبير يتم إطلاق هذا الحدث حيث يخبر المستخدم ما هو الخطأ الذي حدث أثناء تشغيل التطبيق .



لنبدأ مع الطريقة Appliaction_Startup ، في مشروع Silverlight أضف SilverlightControl بحث يكون اسمها friendsPage مثلا ، في الصفحة الجديدة أضف TextBox و أداة Button فقط لكي نحدث اختلافا بسيطا بينها و بين صفحة MainPage . الآن عد للطريقة Application_StartUp وقم بتغيير قيمة الخاصية RootVisual من new mainpage إلى new FrindesPage الآن قم بتشغيل التطبيق و انظر من أين بدأ التنفيذ ، قبل أن تغلق النافذة أريدك أن تلقي نظرة على شريط العنوان في المتصفح ، ماذا تلاحظ !! لا يوجد شيء يشير بأنك في الصفحة friendsPage ! السبب في ذلك بأن صفحات Silverlgiht لا تتعامل مع <a href> لتنتقل بين الصفحات سنرى ذلك في النقطة القادمة .

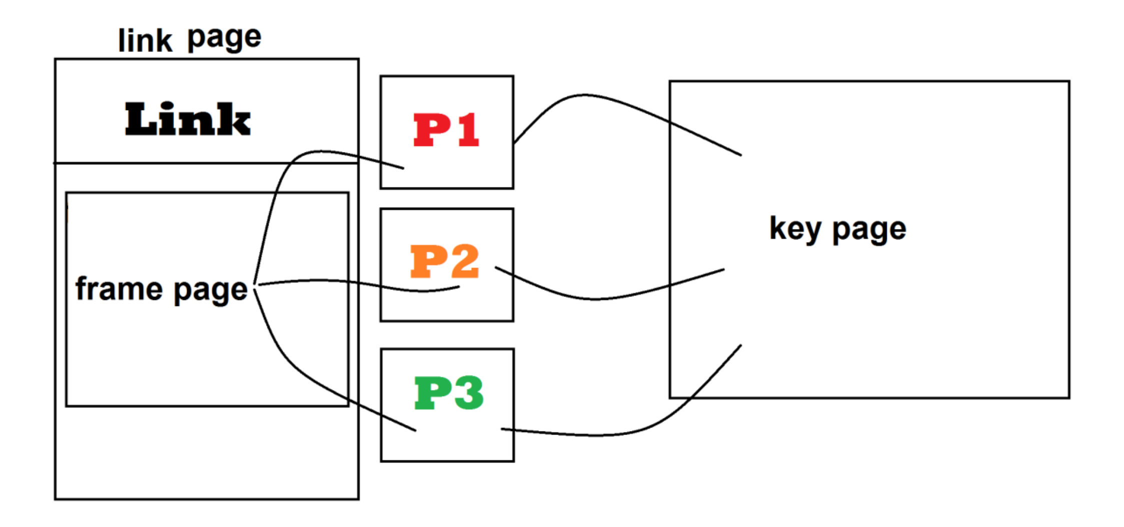
الحدث الآخر هو Application_Exit وهو يقع بعد إغلاق المتصفح ، لنكتب شيء من هذا القبيل :

```
1: private void Application_Exit(object sender, EventArgs e)
2:    {
3:     MessageBox.Show(" did you close the Browser  !");
4:    }
```



الانتقال بين الصفحات في Silverlight

الانتقال بين الصفحات في تطبيقات Silverlight يختلف كثيرا عن تطبيقات الويب العادية ، هناك أسلوبين في التنقل بين الصفحات انظر الى الصورة :



هذه الصورة تلخص كل شيء تقريبا و في كلا الحالتين نحن لدينا صفحة واحدة تقوم باحتواء صفحات أخرى ، في الأسلوب الأول Key page لدينا صفحة فارغة تماما و يتم احتواء الصفحات الأخرى بداخلها ، أما في الأسلوب الثاني link page لدينا صفحة بداخلها أداة frame هذه الأداة بدورها تحتوي على صفحات أخرى .



سنتحدث هنا عن الأسلوب الأول وهو key page و سنرى لاحقا في هذا الجزء الأسلوب الآخر للبدء بالعمل أضف صفحة جديدة تحت مسمى keyPages ثم قم بتعديل شفرة app.xaml بحيث يبدأ التطبيق من هذه الصفحة ، في الصفحة mainpage أضف Button حيث سنستخدمه للانتقال إلى صفحة friendsPage حاول أيضا تغيير التصميم قليلا حتى تشعر بالفرق عند الانتقال .

الآن عد إلى الصفحة KeyPages من شفرة XAML تأكد تماما من أنك قمت بحذف grid الرئيسية المسماة layout root ، ثم ادخل على شفرة سي شارب الخاصة بهذه الصفحة ، أضف في مشيد الرئيسية الشفرة التالية :

```
1: if (this.Content == null)
2:   {
3:    MainPage mainPage = new MainPage();
4:    this.Content = mainPage;
5:   }
```

أيضا قم بإضافة الطريقة GoToThisPage إلى الصفحة keyPages بهذا الشكل :




```
1: public void GoToThisPage(UserControl p)
2:   {
3:    this.Content = p;
4:   }
```

الآن انتقل إلى الصفحة mainPage و انشأ الحدث Click لزر الانتقال لصفحة الأصدقاء ، بداخل هذا الحدث اكتب الشفرة التالية :

```
1: KeyPages kp = this.Parent as KeyPages;
2:  kp.GoToThisPage(new friendsPage());
```

في الصفحة الأخرى اكتب نفس الشفرة لكن مع تغيير القيمة التي تم تمريرها إلى GoToThisPage إلى Mainpage ، لا أعتقد أبدا أن الشفرة تحتاج إلى توضيح فهي أبسط مما يمكن شرحه ! ، قم بتشغيل التطبيق الآن و جرب الأمر .

بهذا نكون قد انتهينا من هذا الفصل تطرقنا فيه لأهم المجلدات و الملفات في مشاريع Silverlight كما رأينا كيف أن ملف app.xaml مهم جدا لعمل تطبيقات Silverlight ، أخيرا رأينا أحد الأساليب التي تستخدم للانتقال بين الصفحات في Silverlight .



الفصل الثامن :

العمل خارج المتصفح

رأينا في مقدمة الكتاب كيف يمكنك تشغيل تطبيقات Silverlight ، خارج المتصفح بشكل سريع في هذا الفصل سنتعرف أكثر على هذا الأسلوب و سنرى كيف يتم تحديث التطبيق من خارج المتصفح ، سنرى أيضا كيف يمكنك الحصول على صلاحيات أكثر من المستخدم عند العمل خارج المتصفح .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- تثبيت تطبيقك خارج المتصفح .
- تخصيص عملية التثبيت .
- ابقاء المستخدم على أخر التحديثات في التطبيق .
- الحصول على صلاحيات أكبر من المستخدم .



تثبيت تطبيقك خارج المتصفح .

ابدأ مشروع جديد في Silverlight ، في منتصف الشاشة الرئيسية أضف TextBox واكتب بداخله Project out of Browsers ، حدد الصفحة mainPage واضغط بزر الفأرة الأيمن و اختر Open with Expression Blend ، في Blend و من القائمة Project حدد الخيار Silverlight Project setting ثم اختر Enable Application Out of Browser ، ثم اضغط على CTRL + Shift + B لبناء التطبيق ثم CTRL + Shift + S لحفظ هذه الإعدادات .

أغلق Blend ثم عد إلى Visual Studio ، قم بتشغيل التطبيق الآن لن تلاحظ شيء ، و لكن انقر على زر الفأرة الأيمن ستجد خيارا جديدا بعنوان Install Out of browsers project in this computer ، انقر هذا الخيار ستخرج لك نافذة التثبيت و هي النافذة التي سيراها المستخدم حدد Start menu و الخيار desktop ثم انقر OK، سيتم تثبيت البرنامج خارج المتصفح و سترى أيقونة التطبيق على سطح المكتب و في قائمة ابدأ ، و سترى التطبيق أيضا في قائمة البرامج في لوحة التحكم.





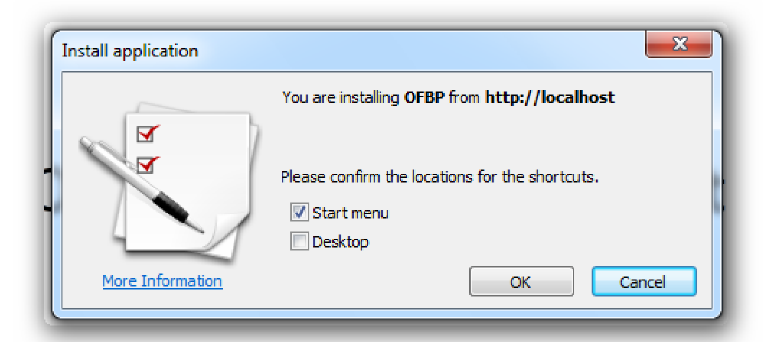
اذا قمت بتشغيل التطبيق سواء من المتصفح أو من سطح المكتب ستلاحظ تغيير الخيار install إلى remove ، تخصيص عملية التثبيت ليست صعبة ، بحكم أننا لم نتحدث حتى هذه اللحظة عن عملية التحديث فأرجو أن تقوم بإزالة التطبيق و حتى تقوم أيضا بتجربة الأمر remove .



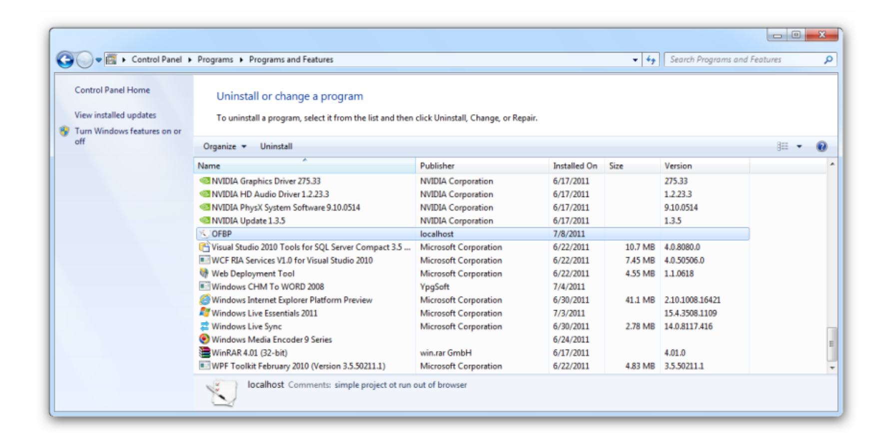
تخصيص عملية التثبيت .

عد إلى visual studio و حدد مشروع Silverlight ثم اختر الأمر Properties حدد التبويب Silverlight أسفل التبويب اضغط على الزر out of browser setting ستجد نافذة إعدادات التطبيق ، من هنا يمكنك تعديل العديد من الخيارات ، مثل اسم الشاشة الرئيسية و عرض النافذة ، و اسم مختصر للتطبيق و أيضا تفصيل التطبيق الذي سيعرض في قائمة البرامج في لوحة التحكم وأيضا أيقونة البرنامج ، وشكل إطار التطبيق في الأسفل ، هناك أسلوب متبع عند التعامل مع الأيقونات ، في البداية يجب أن تكون الأيقونات من النوع png و ليس icon ، ثانيا يجب أن تكون داخل مجلد التطبيق ، الأسلوب المتبع هنا هو إنشاء مجلد جديد في المشروع بعنوان icon ووضع الأيقونات فيه ، عدل خيار التطبيق كما يحلو لك و لكن لا تقوم بتحديد الخيار Require elevated trust when running outside the browser ، قم ببناء التطبيق من جديد سواء باستخدام F6 أو CTRL + Shift + B أو من قائمة Build ، ثم قم بتشغيل التطبيق و حاول التثبيت و انظر إلى الأيقونة الجديدة للتطبيق .





كما تلاحظ تغيرت الأيقونة و تغير أيضا معها اسم المشروع قم بتثبيت التطبيق على سطح المكتب، من قائمة ابدأ انتقل إلى لوحة التحكم ثم البرامج ثم إزالة البرامج و لاحظ التطبيق وقد تم تثبيته على جهاز المستخدم ، كما في الصورة :



لننتقل الآن للخطوة التالية و هي عملية تحديث التطبيق ، أو إن صح التعبير إبقاء المستخدم مطلع على أخر التحديثات .



ما زالت لدينا مشكلة هنا ، كيف يعرف المستخدم أنه يمكنه تثبيت البرنامج على جهازه! لذلك يتطلب عليك إخباره بهذه الأمر ، لعمل ذلك سنضيف زر إلى الصفحة الرئيسية سنستخدم هذا الزر لعملية التثبيت انشأ حدث Click للأداة Button ، لتنفيذ هذه الفكرة يجب أولا أن نتأكد بأن التطبيق ليس مثبتا على جهاز العميل و بالتالي عرض النص install أما إذا كان مثبتا سنخفي الزر كليا ، في المشيد الرئيسي للصفحة mainpage سنتحقق من أن المستخدم لم يقم بتثبيت التطبيق من قبل بالشكل التالي :

```
1: if (Application.Current.InstallState != InstallState.Installed)
2:   {
3:  button1.Content = "Install";
4:   }
5:  else
6:  {
7: button1.Visibility = System.Windows.Visibility.Collapsed;
8: }
```

أما في الحدث Click فسنكتب أمر التثبيت بهذا الشكل :

```
1: Application.Current.Install();
```

فقط ! قم ببناء التطبيق الآن و جرب إزالته ثم شغل التطبيق من جديد و جرب الزر .



ابقاء المستخدم على أخر التحديثات في التطبيق

أحد أهم الأمور التي كانت تشكل أزمة نووية تقريبا لدى مطوري التطبيقات هي عملية تحديث التطبيقات التي يقومون بتطويرها لدى المستخدم و إبقاء المستخدم على أخر التحديثات ، لمعالجة هذه المشكلة كان هناك العديد من الحلول ، من أشهر هذه الحلول هو كتابة ملف على الخادم و قراءته و بتغيير رقم الإصدار في الملف عن رقم الإصدار المثبت تتم عملية التحديث .

في Silverlight الأمر مختلف تقريبا كليا ، هنا لا يوجد ملف بل أن أدنى تغيير في تصميم الصفحات أو إضافة ملفات جديدة يطلق مباشرة الحدث Update ، للأسف لا تتم العلمية بشكل تلقائي بل يجب عليك تفعيلها يدويا ، لعمل ذلك في البداية سنتحاج للتأكد من وجود تحديث في المشيد الرئيسي للصفحة ، أضف الشفرة التالية :

```
1: Application.Current.CheckAndDownloadUpdateAsync();
```

أعتقد أنك قد أصبحت تعرف كيف تتعامل مع الطرق Async من خلال الفصول السابقة فهذه الطرق تطلق حدث Completed ، اكتب :

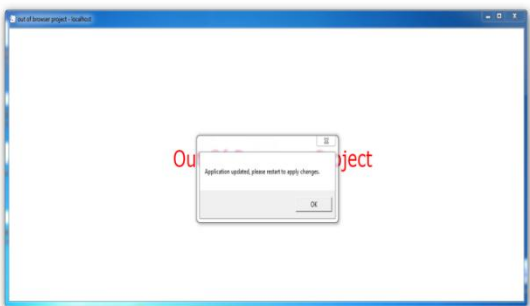
```
1: Application.Current.CheckAndDownloadUpdateCompleted+=
```

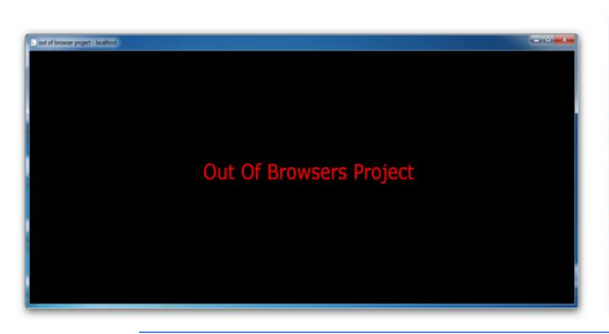


و انقر على زر tab مرتين من لوحة المفاتيح حتى يتم إنشاء الحدث بشكل تلقائي داخل هذا الحدث سنخبر المستخدم مثلا إذا كانت هناك تحديثات جديدة أم لا و ذلك باستخدام المتغير e ، الشفرة :

```
1: void Current_CheckAndDownloadUpdateCompleted(object sender,
2: CheckAndDownloadUpdateCompletedEventArgs e)
3:   {
4:  if (e.UpdateAvailable)
5:      {
6:  MessageBox.Show("Application updated, please restart to apply
7:  changes.");
8:      }
9:   }
```

للتجربة قم بتغيير لون خلفية الـ grid الرئيسية للأسود ثم قم ببناء التطبيق ، عد إلى أيقونة التطبيق المثبتة على سطح المكتب و قم بتشغيل التطبيق (لاحظ النسخة المثبتة على سطح المكتب يجب أن تحتوي على الطريقة CheckUpdate قبل تغيير لون الـ grid) ، سيخبرك التطبيق بوجود تحديثات ، لتنفيذ هذه التحديثات عليك إعادة تشغيل التطبيق ، أغلق التطبيق و سترى الشاشة السوداء أمامك الآن .



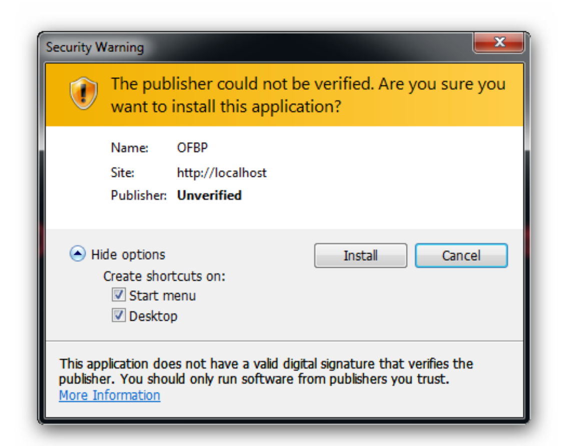




نقطة أخرى في التطبيقات الكبيرة و التي تحتوي على العديد من النماذج لابد أن تكون عملية التأكد من التحديثات داخل الملف app.xaml و بالتحديد في الحدث Application_Startup .

الحصول على صلاحيات أكبر من المستخدم

رأينا في فصل " تخزين البيانات على جهاز المستخدم " كيف أنك محدود كليا بحجم ٢ ميجا ، إذا أردت مثلا تطوير تطبيق يقوم بتسجل ملفات فيديو مثلا فأنت بحاجة إلى مساحة أكبر من ٢ ميجا ، للحصول على صلاحيات أكبر يمكنك فقط تحديد الخيار Require elevated trust when running outside the browser من نافذة out of browser setting .



ستتغير نافذة التثبيت عندها و ستخرج نافذة مشئومة نوعا ما لتخبر المستخدم بتحذير أمني ☹ (انظر الصورة على اليسار).



هناك مشكلة أخرى عند العمل خارج المتصفح ، في التطبيقات التي تحتاج للاتصال بقاعدة البيانات عليك التأكد دائما بوجود اتصال متوفر بالإنترنت كي تكون متصل بقاعدة البيانات لعمل ذلك سنحتاج إلى فضاء الأسماء NetworkInformation التابع لـ System.net سأضيف أداتين من النوع label إلى الصفحة الرئيسية بحيث تعرض حالة الاتصال بالإنترنت للمستخدم ، في المشيد الرئيسي للتطبيق سنطلق حدث NetWorkAccessChanges ، و في هذا الحدث سنعرض إما Connect أو Disconnect . الشفرة :

```
1:  NetworkChange.NetworkAddressChanged += new
2:  NetworkAddressChangedEventHandler(NetworkChange_
3:   NetworkAddressChanged );
```

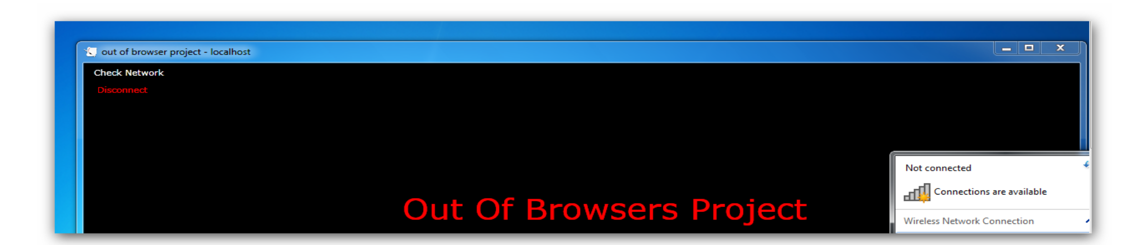
ثم في الحدث NetworkChange_NetworkAddressChanged اكتب التالي :

```
1:  void NetworkChange_NetworkAddressChanged(object sender
2:  , EventArgs e)
3:    {
4:    if (NetworkInterface.GetIsNetworkAvailable())
5:        {
6:      lblCheck.Content = " Connect  ";
7:      lblCheck.Foreground = new SolidColorBrush(Colors.Green);
8:        }
9:         else
10:       {
```




```
11:     lblCheck.Content = " Disconnect  ";
12:     lblCheck.Foreground = new SolidColorBrush(Colors.Red);
13:        }
14:  }
```

قم بتنفيذ التطبيق الآن لن يحدث شيء حتى تقطع الاتصال بالإنترنت ، بمعنى إذا قام المستخدم بتشغيل التطبيق وكانت الشبكة أصلا غير موجودة لن يحدث شيء ، لمعالجة هذه المشكلة يمكنك فقط نسخ محتوى الحدث NetworkChange إلى المشيد الرئيسي في الصفحة mainPage .



إذاً رأينا في هذا الفصل كيف يتم تنفيذ تطبيقات Silverlight خارج المتصفح بشكل كامل تقريبا ، تطرقنا إلى عملية التثبيت و كيف يتم تخصيص عملية التطبيق ثم رأينا كيف تتم علمية تحديث التطبيق بدون استخدام أية ملفات أو ما شابه ، أخيرا رأينا كيفك يمكنك مراقبة الاتصال بالإنترنت في حالة كان التطبيق يحتاج إلى اتصال الانترنت .

٤

الجزء الرابع

التعامل مع الـThemes

وأداة DeepZoom

في هذا الجزء ستتعرف على كيفية التعامل مع Themes ، بالإضافة إلى استخدام أحد أهم الأدوات التي تم تطويرها لـ Silverlight وهي أداة DeepZoom التي تستخدم لعرض الصور .

بعد الانتهاء من هذا الجزء يجب أن يضاف إلى معرفتك التالي:

- التعامل مع الثيمات
- استخدام أداة Deep zoom .



الفصل التاسع :

التعامل مع الـ themes

إذا كنت قادم من تطبيقات ASP.net فمن المؤكد أنك قد تعاملت يوم ما مع ما يسمى بـ CSS و غيرها من التقنيات والأدوات التي تستخدم لإنشاء تصميم معين بحيث يكون هذا التصميم عام على جميع أجزاء المشروع ، هنا يأتي دور themes و دور ملف app.xaml و لكن هذه المرة سنتعامل مع xaml بدلا من شفرة C sharp . أخيرا سنرى كيفية استخدام themes جاهزة بدلا من إنشائه من الصفر .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

- تعيين الـ theme لأداة معينة و جعلها كمصدر في البرنامج باستخدام app.xaml .
- استخدام theme الخاصة بـ Silverlight toolkit .
- استخدام theme جاهزة .



تعيين الـ theme لأداة معينة و جعلها كمصدر في البرنامج باستخدام app.xaml :

لنبدأ بالعمل في هذه النقطة نريد أن نقوم بتغير شكل الأداة button باستخدام Blend ، افتح مشروع جديد في Silverlight ثم افتح الصفحة الرئيسية في Blend ، أضف في الصفحة button أو زر ، حدد Button و انقر بزر الفأرة الأيمن من أسفل القائمة حدد الخيار Edit Template و من ثم حدد الخيار Edit Copy ستظهر أمامك نافذة Create Style Resource في هذه النافذة لديك خياران،

Create Style Resource

Name (Key)

ButtonStyle1

Apply to all

Define in

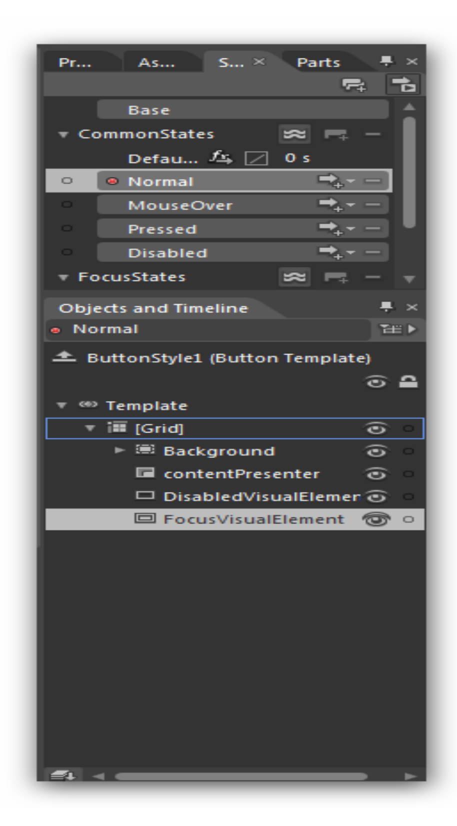
Application

This document — UserControl: <no name>

Resource dictionary

New...

OK — Cancel



إما أن تقوم بتعديل Style لهذا الزر داخل هذه الصفحة فقط أو أن تقوم بتعديل الخيار إلى Application بحيث تستطيع استخدامه على كامل المشروع ، حدد الخيار This document الآن دع اسم الـ style كما هو ثم انقر على موافق .

ستحصل الآن على تحكم كامل بـ style الخاص بهذا الـButton ، ستجد جميع animation في نافذة status كما رأينا سابقا في الفصل الأول ، و سترى جميع الأجزاء التي يتكون منها button داخل OATL لنبدأ الآن بتعديل الـ status الخاص بأداة Button .

حدد Base قم بإجراء بعض التغييرات على الألوان كما رأينا في الفصل الأول ، إذا قمت بتحديد العنصر Background من OATL ستلاحظ علامة صفراء في جزء الألوان في أعلى جزء الخصائص ، هذا يعني أنك لا تستطيع تعديل هذا الجزء من هنا و لكن عليك إعادة بنائه من الصفر ، لذلك انقر على النقطة الصفراء و حدد الخيار rest و ابدأ بإعادة التصميم . لاحظ أن الزر لديه العديد من الحالات لذلك ربما تريد أيضا تعديل كل هذه الحالات .



لن أشرح أي شيء عن طريقة التصميم فقد تعاملنا معها من قبل في الفصل الأول عدل ما تريد و امضي بعض الوقت باستكشاف أجزاء العنصر Button بعد أن تنتهي أناملك الخلابة من التعديلات التي ترغب بها ، اضغط على CTRL + S ، أغلق Templers من أعلى منطقة التصميم .

افتح الصفحة mainPage مجددا من نافذة Projects ستلاحظ عندها بأن style الجديد قد تم تطبيقه على الـ Button الموجود أصلا ، و لكن إن أضفت Button أخر لن ترى أي شيء ! يجب عليك تحديد Miscellaneous Style System.Windows.Style style لهذا العنصر من الخاصية style كل أدوات Silverlight تتمتع بهذه الخاصية حدد الخاصية من شريط properties أو ابحث عنها من أعلى الشريط ، انقر زر المربع بجانب الخاصية ، ثم حدد الخيار local rescores ستجد الـ style الجديد حدد الخيار و سترى هذا style قم بتطبيقه على الزر الجديد .

إذا أين تم حفظ هذا الـ style ، افتح ملف شفرة XAML للصفحة الرئيسية و انظر إلى العنصر الجديد الذي تم اضافته ، و الذي يبدأ بالوسم UserControls. resources .

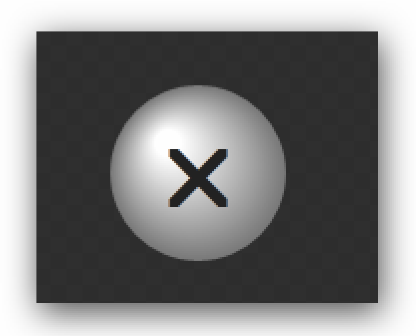
الأن لننتقل إلى الجزء الأخير و هو استخدام الـ Style على مستوى المشروع هذه المرة ، إذا قمت بإضافة صفحة جديدة إلى المشروع و أضفت زر إليها و حاولت أن تفعل نفس الأمر لن تجد أي Style ، لذلك يتم استخدام الملف app.xaml ، بمعنى إذا كنا في البداية عند النافذة create style resources الخيار Application فسوف يتم حفظ الـ style داخل الوسم <Application.Resources> بدلا من



UserContorls ، افتح ملف app.xaml و في شفرة XAML أضف الشفرة التالية إلى عقد <Application.Resources>:

```
1: <Style x:Key="CloseButton" TargetType="Button">
2: <Setter Property="HorizontalAlignment" Value="Right"/>
3: <Setter Property="Width" Value="50"/>
4: <Setter Property="Height" Value="25"/>
5: <Setter Property="Template">
6: <Setter.Value>
7: <ControlTemplate>
8: <Border x:Name="brd1" Width="22" Height="22"
9: CornerRadius="15">
10: <TextBlock x:Name="txt1" Foreground="#222"
11:    TextAlignment="center" Text="r" FontSize="11"
12:    VerticalAlignment="center" FontFamily="Webdings"/>
13: <Border.Background>
14: <RadialGradientBrush GradientOrigin=".3, .3">
15: <GradientStop Color="#FFF" Offset=".15"/>
16: <GradientStop Color="#777" Offset="1"/>
17: </RadialGradientBrush>
18: </Border.Background>
19: </Border>
20: </ControlTemplate>
21: </Setter.Value>
22: </Setter>
23: </Style>
```



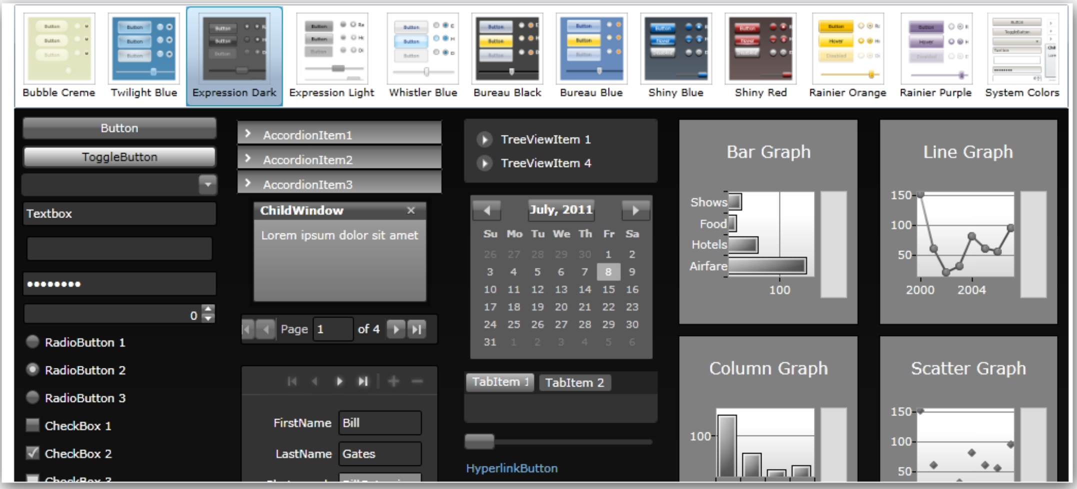


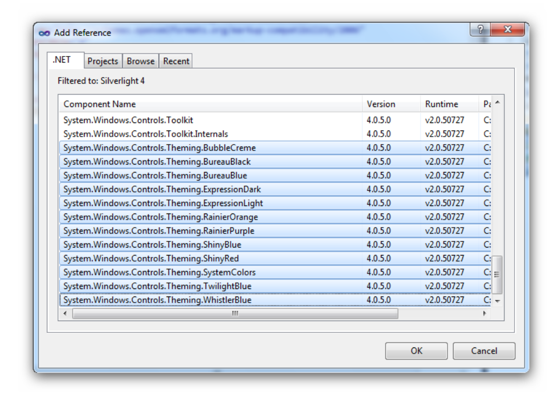
لا تنزعج من شفرة XAMLفقد قمت بعملها باستخدام Blend و لكن أريد أن تحصل على نفس الـ style الذي قمت أنا بإنشائه ، الآن عد مجددا إلى اداة Button في الصفحة الثانية و افعل الخطوات ذاتها عند تغيير الـ style لأداة Button الأولى هذه المرة ستجد الخيار CloseButton ، يمكنك أيضا استخدام هذا الـ style في أي جزء من أجزاء المشروع .

استخدام theme الخاصة بـ Silverlight toolkit

أتمنى أن تكون قد تابعت الكتاب من المقدمة ، لقد طلبت منك في المقدمة أن تقوم بتثبيت أداوت toolkit الخاصة بـ Silverlight،تحتوي هذه المكتبة على العديد من themes الجاهزة للاستخدام، افتح مشروع جديد في Silverlight ، قم بتشغيل الأمثلة لأداة Toolkit وذلك من قائمة ابدأ في مجلد Microsoft Silverlight 4 Toolkit April 2010 ابحث عن Toolkit Samples ، ستجد عنصر تحت مسمى theming داخل هذ العنصر هناك تقريبا ١٢ themes ، سنأتي الآن لطريقة استخدام هذه themes داخل مشاريعك في Silverlight .







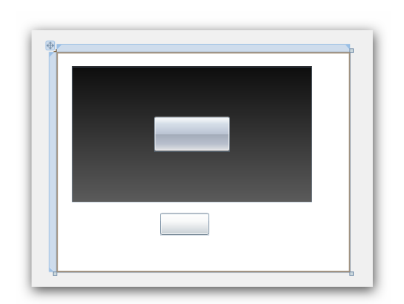
لإضافة themes عليك أولا اضافة المراجع الخاصة بها ، في مجلد references أضف المكتبات كما في الصورة على اليسار بعد اضافة المكتبات الخاصة بـ theming نستطيع الآن أن نستخدمها من داخل mainpage بالشكل التالي :

```
1:  xmlns:BlackTheming="clrnamespace:System.Windows.
2:  Controls.Theming;assembly=System.Windows.Controls.
3:  Theming.BureauBlack"
```



بعد استدعاء فضاء الأسماء لـ BureauBlack Theme يمكنك بكل بساطة تطبيقه على النموذج باستخدام الحاويات ، حيث أن كل Theme يمتلك هذه الميزة ، داخل الـ Grid و في XAML اكتب :

```
1: <Canvas>
2: <BlackTheming:BureauBlackTheme>
3:
4: </BlackTheming:BureauBlackTheme>
5: </Canvas>
```



هذه هي الحاوية الرئيسية لـ Theme الآن كل شيء ستكتبه داخل هذا الحاوي سيتملك بشكل تلقائي خصائص هذا الـTheme لاحظ أنه يمكنك أيضا الكتابة خارج هذا الحاوي ، في الصورة التي على اليمين لدينا زرين أحدهما داخل الحاوي و الآخر خارج الحاوي ، يمكنك تجربة Themes أخرى من خلال تغيير فضاء الأسماء إلى الـTheme الذي ترغب .



استخدام theme جاهزة

رأينا في الفقرتين السابقتين كيف تقوم بإنشاء style خاص بك و استخدمناه على مستوى التطبيق أو على مستوى الصفحة ، في هذه النقطة سنرى كيف تقوم بإنشاء theme خاص بك و استخدامه في تطبيقات أخرى ، و استخدام Themes أخرى داخل مشروعك .

لنبدأ بالنقطة الأولى ، افتح مشروع Silverlight جديد ، ثم انشأ مجلد اسمه Assets حيث جرت العادة على أن ملفات themes توضع في هذا المجلد ، داخل المجلد أضف عنصر جديد من النوع Silverlight Resource Dictionary سمي الملف مثلا style .

عند إعداد الثيمات هناك بعض المسميات المتفق عليها أولا Corestyles و يستخدم لتعديل ثيمات المكونات الرئيسية لـ Silverlight ، ثم toolkitstyles و هنا تضع ثيمات الأدوات Toolkit مثل أداوت التقييم و القوائم وأداة الإكمال التلقائي سنتعرف لاحقا على أدوات toolkit ، ثم لدينا sdkstyles و هي أدوات SDK مثل Data grid و DataPager و ChildWindow ، ثم لدينا fonts و فيه يتم إعلان الخطوط المستخدمة في التطبيق ، ثم لدينا brushes و يستخدم لتخزين الأشكال و الألوان التي تستخدمها في تصميم الـTheme ، أخيرا لدينا styles و هذا الأخير تقوم به في تخزين أية إجراءات تقوم بتعديل شكل الأداة كما رأينا مع أداة Button سابقا .



سأستخدم الشفرة التي في الأعلى (شفرة CloseButton) لتغيير شكل أدوات Button ، فقط انسخ الشفرة و الصقها داخل الوسم ResourceDictionary ، الخطوة التالية هي إخبار التطبيق بأن لدينا Style مخصص للـButtons داخل ملف app.xaml ، بهذا الشكل :

```
1: <Application.Resources>
2:   <ResourceDictionary>
3:     <ResourceDictionary.MergedDictionaries>
4:     <ResourceDictionary Source="Assets/Style.xaml"/>
5:     </ResourceDictionary.MergedDictionaries>
6:   </ResourceDictionary>
7: </Application.Resources>
```

فقط ، الآن انتقل إلى النموذج الرئيسي و أضف زر ثم من الخاصية style للعنصر الجديد ، الآن أصبح لديك ملف style يمكنك إعادة استخدامه في أكثر من تطبيق .



- System.Windows.Browser
- System.Windows.Controls
- System.Windows.Controls.Data
- System.Windows.Controls.Data.DataForm.Toolkit
- System.Windows.Controls.Data.Input
- System.Windows.Controls.Data.Toolkit
- System.Windows.Controls.DataVisualization.Toolkit
- System.Windows.Controls.Input
- System.Windows.Controls.Input.Toolkit
- System.Windows.Controls.Layout.Toolkit
- System.Windows.Controls.Navigation
- System.Windows.Controls.Toolkit
- System.Windows.Controls.Toolkit.Internals
- System.Xml

حسنا ماذا عن الثيمات الجاهزة ؟ ، كما أسلفت ذكره سابقا كل Theme لا بد أن يحتوي على ملف واحد أو أكثر من الملفات التي في الأعلى، في المرفقات مع المشروع لديك أربعة Themes أكثر من رائعة يمكنك استخدمها مع تطبيقاتك بشكل مباشر حاليا احذف الButton و احذف ملف Style ، افتح المجلد themes في مرفقات المشروع ستجد مجلد بعنوان JetPack داخله مجلد بعنوان assets داخل المجلد الاخير توجد ملفات الTheme ، قم بإضافة جميع الملفات إلى مجلد assets في مشروعك ، إذا حاولت تنفيذ المشروع الآن ستتفاجأ بكم هائل من الأخطاء هذه الأخطاء فقط نتيجة عدم وجود مكتبات Toolkit اللازمة ، حدد المكتبات المطلوبة من نافذة الأخطاء و قم بإضافتها إلى المشروع ، انظر إلى الصورة في الأعلى ، إذا حاولت الآن أن تقوم بإضافة أي عنصر إلى أي صفحة في التطبيق ستجد مباشرة أن هذه الأداة قد حصلت على الـ Them الجديد لها . لا تنسى أن تقوم بإضافة الملفات الأخرى الموجودة داخل مجلد assets إلى الملف app.xaml كما رأيت مع الملف Style.xaml و ليصبح ملف xaml بهذا الشكل :

```
1: <Application.Resources>
2:   <ResourceDictionary>
3:     <ResourceDictionary.MergedDictionaries>
4:     <ResourceDictionary Source="Assets/Style.xaml"/>
```




```
5:      <ResourceDictionary Source="Assets/CoreStyles.xaml"/>
6:      <ResourceDictionary Source="Assets/SDKStyles.xaml"/>
7:   <ResourceDictionary Source="Assets/ToolkitStyles.xaml"/>
8:   </ResourceDictionary.MergedDictionaries>
9:    </ResourceDictionary>
10: </Application.Resources>
```



إلى هنا نكون قد انتهينا من هذا الفصل ، رأينا فيه كيف يتم تعديل Themes الأدوات الجاهزة و كيف يتم استخدام هذه التعديلات على مستوى النموذج أو على مستوى التطبيق بالكامل ، كما رأينا كيف يتم إنشاء Theme خاص بك و استخدامه في تطبيقات أخرى ، تطرقنا للملف app.xaml و رأينا كيف أن هذا الملف يلعب دورا مهما في مشاركة المصادر بين أجزاء المشروع المختلفة ، أخيرا رأينا أجزاء الـThemes المختلفة و رأينا Themes الـ toolkit . في النهاية تصميم تطبيقات Silverlight من الأمور الممتعة جدا عند الحديث عن استخدام Blend و لكنها من الأمور الكئيبة عند استخدام XAML فقط .



الفصل العاشر :

التعامل مع DeepZoom

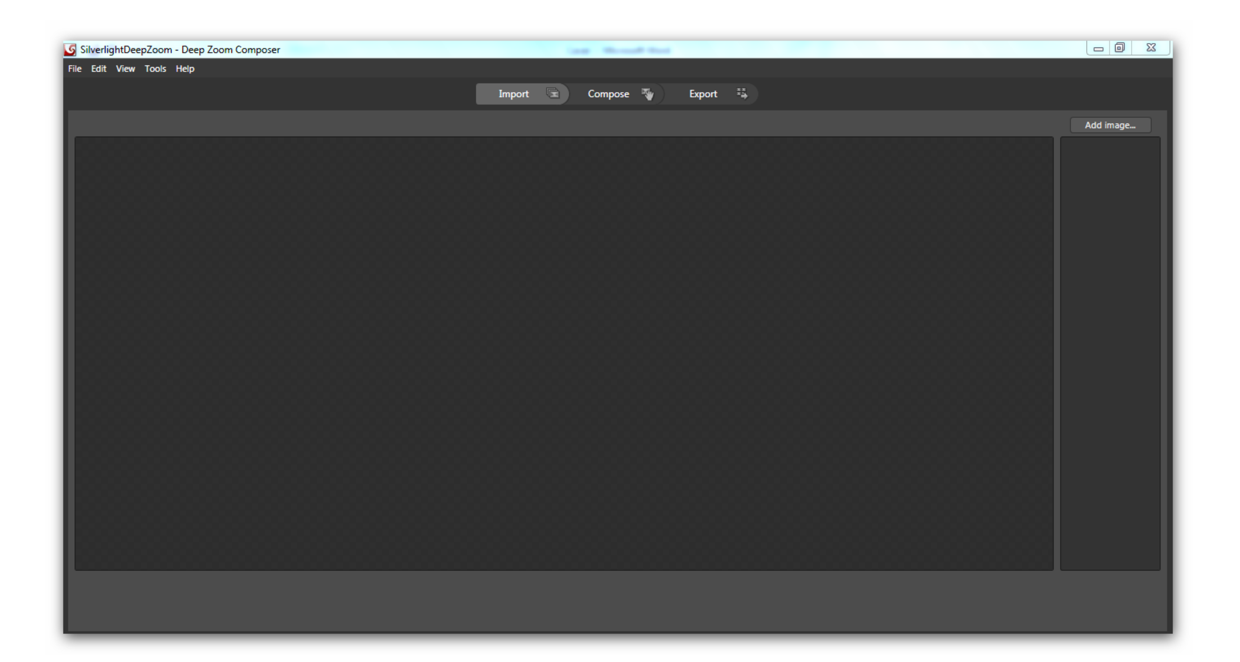
توفر أداة Deep Zoom طريقة جيدة لعرض الصور بدقة عالية و بطريقة أكثر قربا لتطبيقات Silverlight من استخدام أداة Image العادية ، يمكنك رؤية تطبيق مباشر لأداة DoopZoom على موقع hard Rock ، سنرى في هذا الفصل كيف يمكنك استخدام الأداة بدون كتابة سطر واحد من الشفرة ☺ ، و ستعرف حقا معنى أن تصبح مطور Silverlight .

بعد الانتهاء من هذا الفصل يجب أن يضاف إلى معرفتك التالي :

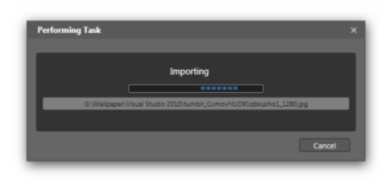
- استخدام أداة DeepZome .

استخدام أداة DeepZoom

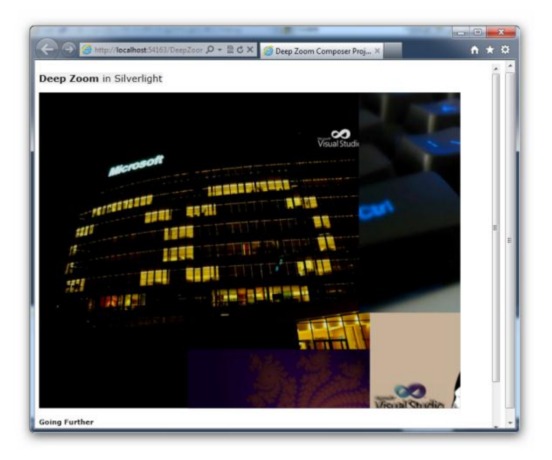
لقد ذكرت في المقدمة رابط الحصول على DeepZoom ، لذلك سأفترض هنا أنك قمت بتثبيت هذه الأداة مسبقا ، من قائمة ابدأ حدد البرنامج Deep Zoom Composer ، قم بإنشاء مشروع جديد و ليكن SilverlightDeepZoom ، ستجد نفسك الآن في بيئة التطوير الخاصة بـ DeepZoom .







اضغط على زر Add Images و أضف الصور التي تريدها ، سيبدء المعالج مباشرة بجلب الصور و جعلها كمصدر للبرنامج (الصورة على اليسار) الآن الخطوة التالية هي إنشاء شكل معين لعرض الصور ، حدد الخيار Compose من الأعلى ، ستجد الآن شريط الصور في الأسفل ، في الجهة اليمنى ستجد شريط layers أو الطبقات ، حيث أن كل صورة تشكل طبقة مستقلة بذاتها ، على اليسار سترى أدوات التنسيق من محاذاة الصور و التكبير و التصغير و خلافه ، أخيرا في منتصف منطقة العمل ، الآن ركز قليلا ابدأ بسحب أي صورة من شريط الصور و رميها على منطقة التصميم ، أثناء عملية الرمي سترى الحدود الحمراء بين كل صورة وأخرى .



بعد أن تحدد الصور التي ترغب بعرضها ، انتقل للخطوة الأخيرة Export ، حدد Export من الأعلى ، سيظهر لك شريط على اليسار، لديك أولا الخيارات output type و منها يمكنك تحديد نوع الملف الذي تريد عمل Export له حيث يمكنك اخراجها على شكل صور أو على شكل تطبيق Silverlight حدد Silverlight ، ثم من خيارات Export Option في الخيار Templates يمكنك تحديد مجموعة من الخيارات ما يهمنا الآن هو الخيار Expression Blend 4 + Source حدد الخيار ، ثم غير اسم المشروع إلى DeepZomeEx اضغط على Export للبدء بعملية التصدير ، انتظر للحظات حتى تنتهي العملية . و لننتقل إلى استخدامها من Blend ، انقر على الخيار Preview in Browser(سيظهر هذا الخيار بعد عملية التصدير) . افتح المشروع الصادر سواء بـ VS أو Blend و ابدأ بتنفيذ المشروع مباشرة ، فقط ☺ .



الخاتمة :

كما تطرقت في بداية الكتيب ، هدفي الرئيسي في هذا الكتيب هو بناء معرفتك حول الأمور الرئيسية في Silverlight ، كبداية في هذا العالم أعتقد أن هذا يكفي ، ما زال هناك المزيد من الأمور المتقدمة لذلك أرجو منك زيارة هذه المصادر لكي تبقى متابعا أولا بأول عما يدور ، و تحصل على آخر الأخبار و الأدوات و الشروحات و خلافه المتعلقة في Silverlight :

- خلاصات المواقع الرئيسية :
  - Silverlight Cream .
  - Silverlight Community .
  - Silverlight Show .
- خلاصات مدونات مطوري Microsoft :
  - Jesse Liberty
  - Tim Heuer
  - Jeff Wilcox
  - Christian Schormann
  - Jaime Rodriguez
  - Mike Taulty
  - Peter Torr
  - Corrina Barber
  - John Papa
  - Pete Brown’s Client Roundup

- خلاصات فيديو و برودكاست :

- Silverlight TV
- Inside Windows Phone
- This Week in Silverlight

- خلاصات المدونات الرسمية :

- Windows Phone Developer Blog
- Silverlight Blogs

- خلاصات أهم مجتمعات Silverlight :

- Dan Wahlin
- Shawn Wildermuth
- Adam Kinney
- Brian Noyes
- Jeremy Likness
- La Liga Silverlight
- Michael Crump
- Victor Gaudioso
- WCF Silverlight team
- Alex van Beek
- Timmy Kokke
- Roboblob



فهرس المحتويات